THE BALLISTICS PROCESSOR OF A MULTIPLE PROCESSOR AIRBORNE TACTICAL SYSTEM.

Harry Andrew Jupin

THESIS

THE BALLISTICS PROCESSOR
OF
A MULTIPLE PROCESSOR AIRBORNE TACTICAL SYSTEM

by

Harry Andrew Jupin

June 1975

Thesis Advisor: U. R. Kodres

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): The Ballistics Processor of A Multiple Processor Airborne Tactical System	5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis; June 1975
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Harry Andrew Jupin	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	12. REPORT DATE: June 1975
	13. NUMBER OF PAGES: 103
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Naval Postgraduate School, Monterey, California 93940	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ballistics processor of a multiple processor airborne tactical system
Three INTEL-8080 microcomputers
General second order Runge-Kutta method of integration of the equations of motion of unguided air-to-surface weapons

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis developed the ballistics processor of a multiple processor airborne tactical system. The multiple processor system consisted of three INTEL-8080 microcomputers: the executive processor, the navigational processor and the ballistics processor. The ballistics processor utilized a general second order Runge-Kutta method of integration of the equations of motion of unguided air-to-surface weapons. The

19. Key Words

Ballistics Trajectory Algorithm

20. Abstract

ballistics processor computed sufficiently accurate and timely solutions to enable the executive processor to extrapolate an accurate release point for the weapon. The algorithm permitted complete flexibility in release conditions and allowed a complete arsenal of air-to-surface weapons currently carried on the A7-E aircraft. The cost of the ballistics processor using current "off the shelf" components is $1635 and the entire tactical system was estimated to cost $3891.

The Ballistics Processor of A Multiple Processor Airborne Tactical System

by

Harry Andrew Jupin
Lieutenant, United States Navy
B.S., Muskingum College, 1968

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1975

ABSTRACT

This thesis developed the ballistics processor of a multiple processor airborne tactical system. The multiple processor system consisted of three INTEL-8080 microcomputers: the executive processor, the navigational processor and the ballistics processor. The ballistics processor utilized a general second order Runge-Kutta method of integration of the equations of motion of unguided air-to-surface weapons. The ballistics processor computed sufficiently accurate and timely solutions to enable the executive processor to extrapolate an accurate release point for the weapon. The algorithm permitted complete flexibility in release conditions and allowed a complete arsenal of air-to-surface weapons currently carried on the A7-E aircraft. The cost of the ballistics processor using current "off the shelf" components is $1635 and the entire tactical system was estimated to cost $3891.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure

ACKNOWLEDGEMENT

The author wishes to express his thanks to Associate Professor Kodres for his overall assistance and guidance throughout the project, and his keen insight into micro-computer programming. Special thanks go out to my family, Jeanne and Tim, who have constantly served as an inspiration and stabilizing influence throughout the preparation of this thesis.

I. INTRODUCTION

Military airborne tactical systems are used to aid the pilot and bombardier/navigator in their specific functions. For an attack aircraft, the tactical system is used in the following manner:

1. to navigate to a given target
2. to issue steering commands enroute and/or to guide the autopilot
3. to display to the crew the current target location in relation to the aircraft
4. to solve ballistics equations which enable the crew to release a weapon at the appropriate instant.

The presently operational systems employ a small general purpose computer, such as the IBM 4 PI system. In this thesis, the ballistics processor of a multiple processor organization of a typical tactical system for an attack aircraft is presented. The system's architecture made use of three INTEL-8080 microcomputers, each of which was dedicated full time to its assigned function. The purpose of the multiple processor organization is to exploit the presently emerging inexpensive microprocessor technology to solve military tactical problems. The system's organization is shown in Figure 1. Because microcomputers presently have no hardware multiply-and-divide functions, the programmed software multiply-and-divide functions caused the microcomputers

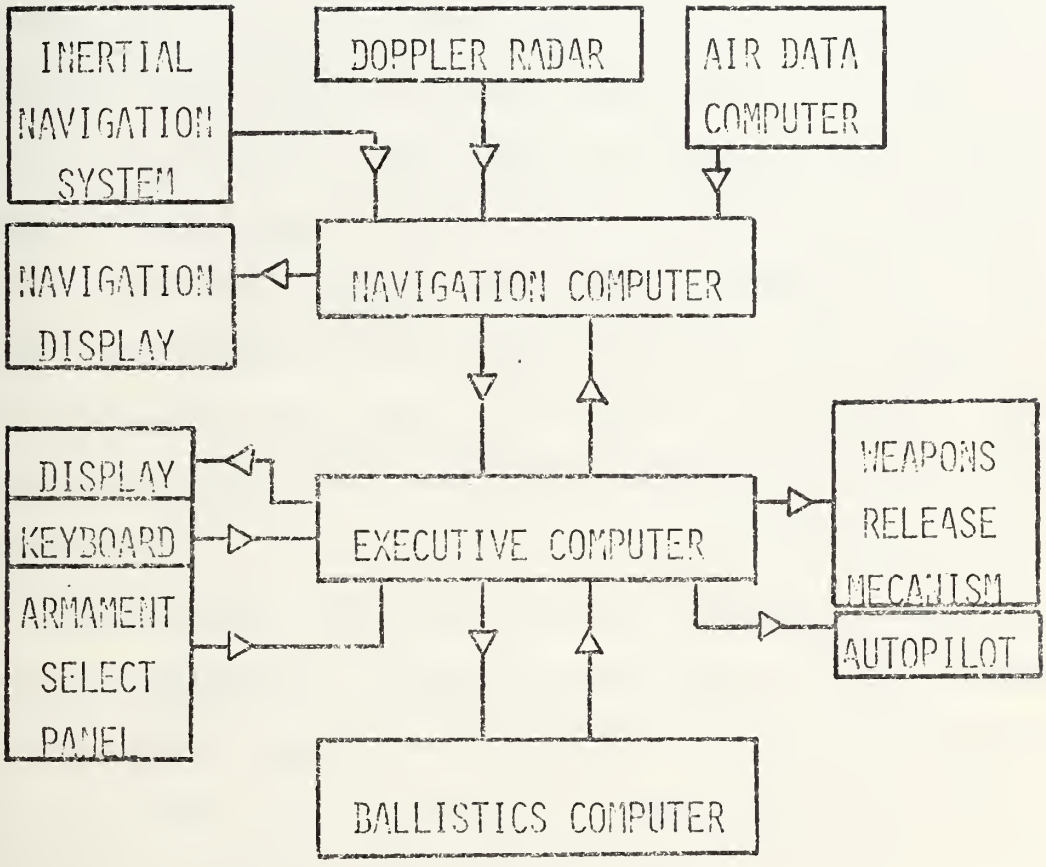


FIGURE 1. ATTACK AIRCRAFT'S
TACTICAL SYSTEM'S INTERCONNECTION

to run on a typical mix of operations approximately six times slower than the presently operational computer systems. By reorganization of the computations and the use of table look-up techniques, the described system was capable of meeting the real-time requirements of an airborne weapon delivery system.

The ballistics computer uses the best estimate of the position, dive angle, altitude, and velocity of the aircraft in its computation. It is also provided with the weapon identification (for the particular drag characteristics) and the wind velocities. From this information and the target's altitude (above Mean Sea Level), the computer determines the down range travel and time of fall of the unguided weapon (i.e., the impact coordinates in the target plane).

The accuracy with which a ballistic weapon can be delivered against a target depends greatly upon the accuracy of the sensor-supplied release parameters as well as the accuracy with which the ballistics equations are solved to predict the weapon's down range travel.

Most aircraft unguided air-to-ground weapons can be described as ballistic projectiles. The only forces acting on them after release from the aircraft are gravity and aerodynamic drag. Bullets, streamlined bombs, drogued (retarded) bombs, cluster munitions and unguided rockets (after motor burnout) are all ballistic projectiles. Guided

weapons and weapons developing lift are not ballistic projectiles nor are they considered in this development.

This thesis first describes the over-all tactical system in order to give the reader a perspective of how the ballistics processor functions within the entire weapon system. Included in this background material is a discussion of the navigational computer, the ballistics computer, the executive computer and the intercomputer communications network.

The basic calculations involved in solving the ballistics problem is described next. The assumptions used in the problem formulation are stated; the derivation of the particular differential equations is presented; and the application of the Runge-Kutta integration method is shown to be suitable for current operational airborne weapon systems. The four general classes of weapons and how the algorithm handles each integration is defined. The unique method of compressing the drag data for all twenty-eight weapons is shown to be a simple curve fitting of the drag coefficients with respect to some standard.

The simulation was initially programmed using FORTRAN for ease of testing and debugging. The important phases in programming were in the DERIV and RUNGE subroutines where a close study of the optimum calculation methods was conducted. The program was tested for accuracy of results by comparison with the published range tables (NAVAIR 01-1C-1T-1).

Upon completion of the FORTRAN implementation, the resultant program was translated into PL/M, a high level

language compatible with the selected microcomputer (INTELLEC 80). The selected test cases were verified correct in the PL/M version of the program.

II. BACKGROUND

In order to give the reader a complete understanding of how the ballistics processor functions, this section is devoted to describing the overall tactical system in which the ballistics processor is an important part. The first section describes the function of the navigational computer, the sensors which provide data to the computer and the algorithm used to compute present position.

The ballistics computer will be completely defined in the remaining sections of this thesis and will only be briefly reviewed here. Using the best estimate of position and velocity at a given instant, together with information of the weapon's drag characteristics, the ballistics computer solves a system of four differential equations and calculates the impact coordinates of the weapon in the target plane.

The executive computer controls the display and generates commands to the autopilot and weapon release mechanism. This computer is interrupted asynchronously as fresh data are developed in the navigational and the ballistics computers. The executive computer extrapolates from the data the appropriate release instant of the weapon.

The intercomputer communication net is described next. The star-like communication net provided a simple and rapid means of information transfer.

A. THE NAVIGATIONAL COMPUTER

This subsystem is present in all tactical systems. In operational systems, a subprogram (module) is periodically executed to update the present position by the change in position in the last time increment.

In this computer, the sensor instruments provide digital data to the computer at specified time intervals, controlled by a clock. There are four sets of sensor instruments which provide the data:

1. The Inertial Navigation System (INS)
2. The Doppler Radar (DR)
3. Air Data (AD)
4. Radar Set (RS).

The INS system provides attitude (roll and elevation) and velocity increments in the x, y, z directions. The DR is a velocity sensor that uses the doppler principle for the continuous measurement of aircraft ground track speed and drift angle. The AD computes outputs of modified corrected static pressure, pressure altitude and mach number from static and pitot tube pressure inputs. The RS provides target azimuth, elevation, and range signals to the navigational computer. A magnetic compass measures the aircraft heading in relation to magnetic north. In order to navigate accurately in this mode the wind speed and direction are determined by using the "sensed" velocities of the aircraft. Altimeter measurements are also sampled in this system to update the present position altitude.

There are four basic modes in which this system operates. The crew has a choice of using both the INS (velocities) and DR outputs to minimize the Schuler oscillations. This is the most reliable and accurate mode. If the crew suspects that either the INS or the DR system is malfunctioning, they may use only the data from one of the sources. The least reliable and inaccurate mode is the last resort, when both the DR and the INS are malfunctioning. The AD mode uses the externally estimated wind vector and true airspeed of the aircraft to determine the position in latitude and longitude.

In all operating modes, the same basic technique of smoothing input data is used. The least squares curve fitting technique is used to fit a quadratic polynomial to the most recent n ($n \leq 16$) data values. Use of the Legendre orthogonal polynomial functions is made in order to make the numerical calculations as accurate and rapid as possible. Smoothing the short time increment data values will generate less total error in the results.

Using the smoothed data values, the appropriate calculations are made to estimate the distance increments in the past time interval. As soon as the new value is generated, the display of the navigational computer is refreshed, the executive computer is interrupted and this most recent estimate is transferred to the executive computer. If the navigational computer is functional, the crew has information on the present position, independent of the executive computer.

The least squares curve fitting technique enables the system to diagnose sensor malfunction quite easily. If the sensor generates fluctuations in the input data above a certain threshold, then the sensor can be diagnosed as malfunctioning. Similarly, if two sensors disagree above an allowable threshold, then an automatic data analysis will suggest which sensor is failing. If the crew concurs in this diagnosis, the operating mode of the navigational system can be changed to exclude the faulty sensor.

B. THE BALLISTICS COMPUTER

The ballistics computer is provided with the present estimates of the position, dive angle, altitude, and velocity of the aircraft. It is also provided with the type of weapon and wind conditions. From this information and the target's altitude, the computer determines the downrange travel of the weapon, if released at this instant. The assumption of no guidance after release from the aircraft is still applicable. However, the weapon may have a rocket assist, or a retarded mode of fall.

C. THE EXECUTIVE COMPUTER

The executive computer controls the main display and generates steering commands to the autopilot and firing pulses to the weapons release mechanisms. One of its critical functions is to extrapolate from the data generated by the ballistics computer the weapon release point.

A real time clock is used to record the time at which the asynchronous data were generated by the ballistics computer. The position of the impact point in the target plane can be described by a quadratic function of time. Therefore, the real time release point (adjusted for time delays) can be extrapolated very accurately from these data without additional extensive calculations.

D. THE INTERCOMPUTER COMMUNICATIONS NET

In order to keep the design of the intercomputer communications net simple and inexpensive, the starlike network was chosen. The executive computer is the center of the star and one input port is multiplexed to the two output ports of the navigational and ballistics computer as shown in Figure 2.

The intercommunication takes place in the following manner: An initializing phase of the program interrogates the keyboard to obtain data about the initial position of the aircraft. The latitude and longitude are entered in degrees and minutes to the nearest 1/10 minute; the altitude is initialized at the height above mean sea level. These data are made available to the navigational computer. Also the type of weapon and the position of the target (if known in advance) is made available to the ballistics computer. Similarly, the mode of navigation is determined (if possible prior to flight) and relayed to the navigational computer.

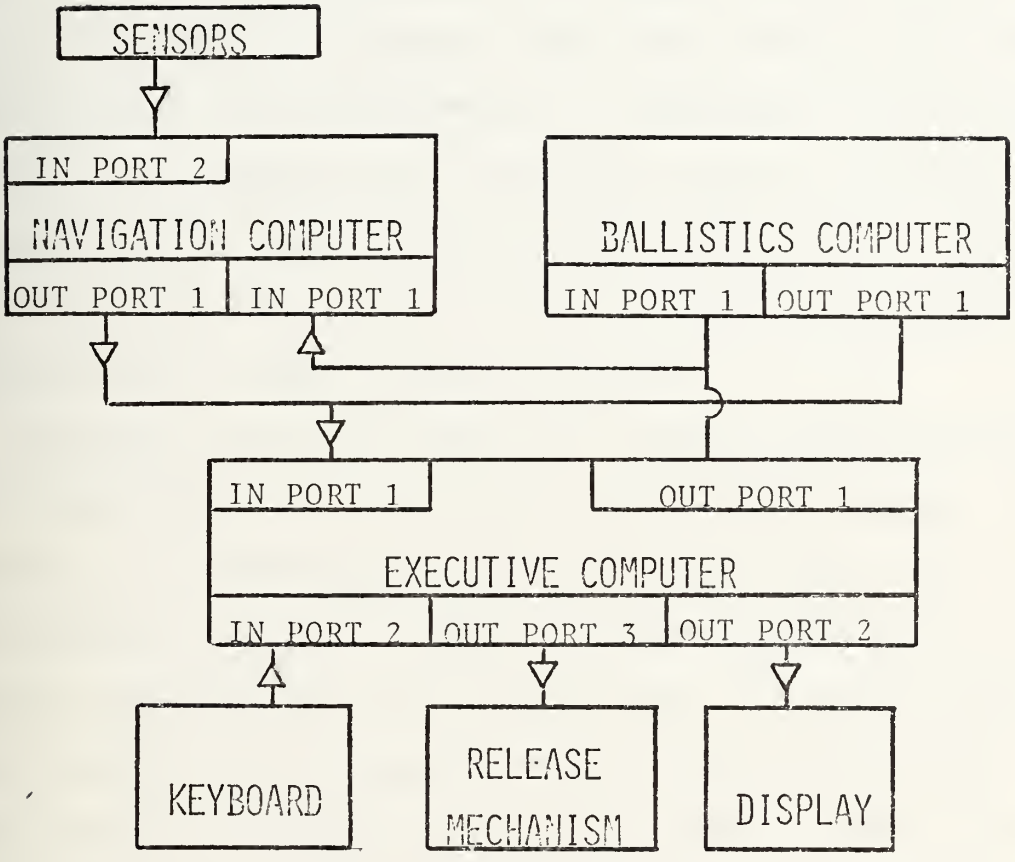


FIGURE 2. THE INTERCOMPUTER COMMUNICATION NET

The execution phase of the computer program makes use of the real time clock in the executive computer which is asynchronously interruping the navigational computer in order that sensor information be transferred via input port 2 to the navigational computer. When the navigational computer has completed its calculations, it interrupts the executive computer and transfers the calculated current position sequentially via input port 1 to the main computer. The amount of information sent is 9 bytes, and the transfer is accomplished in about 250 micro-seconds.

Similarly, when the ballistics computer has completed its calculations, it interrupts the executive computer and transfers its information sequentially via input port 1 to the executive computer. Only 6 bytes of information are sent in approximately 170 micro-seconds. Between interruptions, the executive computer updates the display using output port 2. Calculations are continuously made to extrapolate the release instant of the weapon, and the release mechanism is activated when appropriate via output port 3.

III. BALLISTICS PROBLEM

The equations of motion governing the trajectory of a ballistic projectile are a simple-appearing set of second-order differential equations whenever the drag is neglected. However, when a reasonably accurate model of the aerodynamic drag caused by the projectile's motion through the air mass is included in these equations they are rendered nonlinear, and no general solution in closed form has been found that is satisfactory for solution by an airborne fire-control computer.

A. BASIC ASSUMPTIONS

The mathematical model was chosen to represent reality as accurately as possible. The armed forces publish range tables for various weapons and therefore the same model was chosen as the one used to construct these tables (NAVAIR 01-1C-1T-1).

The equations of motion were developed assuming the projectile is a point mass acted on only by the force of gravity and the retardation forces due to air resistance. The trajectory was restricted to a plane by ignoring the crosstrail effects of the wind. These crosstrail effects and the component of wind which affects drag are considered in the executive computer.

The assumptions are summarized below:

1. The Earth is flat and nonrotating.
2. The gravitational attraction is constant.
3. The projectile is a point mass.
4. The projectile is not powered (except rockets) and has a constant mass.

B. DERIVATION OF THE EQUATIONS

The weapon's fall is assumed to be governed by a drag force which is proportional to the square of the velocity. Newton's equation of motion becomes

$$m\, dVt/dt = -\, g\, m - \|Vt\|^2\, Cw\, Vt/\|Vt\|$$

where the only two forces acting on the weapon are gravity and aerodynamic drag.

Vt - total velocity

m - mass of the projectile

g - force due to the Earth's gravitation

Cw - drag coefficient which depends on the characteristics of the weapon and may vary with time (in case the weapon has two drag curves), and it varies slightly with airspeed.

In order to solve this differential equation, initial conditions, that is the position and velocity, must be specified. The most convenient coordinate system is height "y" above sea level and "x" downrange travel, which is zero at the time of release.

In the x and y component form, Newton's equation becomes

$$dVx/dt = - \|Vt\| /m * Vx * Cw$$

$$dVy/dt = - g - \|Vt\| /m * Vy * Cw$$

$$dx/dt = Vx$$

$$Dy/dt = Vy$$

$x(0) = 0$ downrange travel

$y(0) =$ present altitude

$Vs(0) =$ present horizontal velocity

$Vy(0) =$ present vertical velocity

C. RUNGE-KUTTA APPLICATION

The choice of the "best" Runge-Kutta (R-K) formula for a particular problem was not entirely obvious. The lower order formulae are simple, but require more integration steps than the more accurate higher order formulae. On the other hand, the complexity of the higher order formulae may require so much computer time that this becomes the overriding consideration. Generally, it was found to be safer to use the lower order formulae when discontinuities in the coefficients of the differential equations can be expected. Such discontinuities frequently occur in bomb trajectories (e.g., drogue deployment or thrust termination). The second-order R-K formula also afforded the greatest flexibility in the distribution and size of the integration steps. These considerations, along with their simplicity, led to the choice of the second-order R-K method as the most appropriate for the ballistics algorithm.

Let $U1 = Vx$ $\quad F1 = -1/m * \|Vt\| * Vx * Cw$

$U2 = y$ $\quad F2 = Vy$

$U3 = Vy$ $\quad F4 = -1/m * \|Vt\| * Vy - Cw - g$

The differential equations may be expressed as

$$dUi/dt = Fi(U1,U2,U3,U4) \qquad i = 1,2,3,4$$

Therefore, the Runge-Kutta second order method used in the program becomes

$$m1i = \Delta t * Fi(U1,U2,U3,U4) \qquad i = 1,2,3,4$$

$$m2i = \Delta t * Fi(U1 + c\Delta tm11, U2 + c\Delta tm12, U3 + c\Delta tm13, U4 + c\Delta tm14) \qquad c = 0.7,\ i = 1,2,3,4$$

$$U'i = Ui + 1/2c * ((2c - 1) * m1i + m2i) \qquad i = 1,2,3,4$$

The procedure computes new values for the variables until the remaining time of fall, DTV, (as computed for a vacuum release), is less than the value of the next integration time increment. The procedure then uses the value of DTV for the final integration step size, and in the calculation of the down range travel of the weapon.

D. CLASSES OF WEAPONS

1. Streamlined Bombs and Bullets

The process described above represents streamlined bombs quite well. After being fired, a bullet is really just a small, streamlined bomb and was handled as such. The muzzle velocity (3300 feet per second) of a particular gun-bullet combination was added to the aircraft velocity.

2. Drogued Bombs

These bombs are more complex in their trajectory modeling. Their common characteristic is that they are all released in a relatively low-drag configuration. At some time after release, they deploy vanes or a parachute which greatly increases the total drag. To compensate for this change of drag characteristics, the initial integration step size was chosen to coincide with the deployment of the high drag devices. Thereafter, there was no difficulty in calculating the trajectory of the drogued weapons.

3. Cluster Bombs

These weapons are released as large, low-drag containers. At a predetermined point in their trajectory, which is time dependent, they use some method to dispense smaller weapons. These smaller weapons are of higher drag than the container. The ballistics computer will predict the impact point of the center of the pattern rather than individual positions of the bomblets. The method of applying the algorithm is to calculate the trajectory of the container to the point where it dispenses the small weapons; switch to a different drag curve which describes the motion of the pattern center; and integrate down to the target altitude. As in the retarded bombs, the initial integration step was chosen to coincide with the act of dispensing the small weapons.

4. Unguided Rockets

Rockets present several unique complexities. They change weight during flight; they have thrust as well as

drag; and they slew around after firing, because their launches may not line up with the aircraft direction of flight. The algorithm treats unguided rockets like a cluster bomb with the first stage having an average thrust as well as an average thrust time. This provides sufficient accuracy and saves computer storage that would be taken storing the thrust profile of each rocket.

E. COMPRESSION OF DRAG DATA

The behavior of a particular weapon's drag coefficient, Cd, as a function of Mach number must be known to calculate impact ranges for that weapon. Normally, this information is given in tabular form with some weapons having more than 100 points in the table. Most aircraft have the capability to carry many different types of weapons. It is desirable to store all the Cd information of an aircraft's weapon repertoire in the airborne computer to avoid loading the Cd data for each different combination of weapons. This requires storing drag coefficient data in a more efficient form than a table.

The following factors influence the choice of a scheme to approximate the drag coefficients for a given set of weapons:

1. Computer storage
2. Computer computation time
3. Delivery envelope of each weapon
4. Allowable downrange and time-of-fall errors.

It would be ideal to use one general, fairly powerful expression to fit all the drag tables for the weapon repertoire of a given aircraft. This is quite difficult to accomplish because weapons like guns and rockets require Cd values for Mach numbers much higher than free-fall bombs. As a consequence of this generally acceptable tabling procedure, extra storage is spent on weapons that do not require the extra capability.

A significant saving can be realized when it is recognized that several free-fall weapons have drag coefficient curves that differ only by a multiplicative factor. The Mk 80 low-drag series are one such group of weapons with the Mk 84 drag curve conventially taken as the reference. Similarly, the Mk 106, CBU bomblet, and Sadeye bomblet are another group with the Garve drag curve as the reference. The specific region of interest of many more drag curves can be approximated by multiplying these two reference drag curves by a factor and then translating them along the Cd and M axes until they match the original drag curves.

To further aid in the explanation of approximating weapon drag curves by this method, let Cd(M) and M be the drag coefficient and Mach number of a weapon drag curve. Let $\underline{Cd}(M)$ be the drag coefficient of a reference drag curve and let $\underline{A}$, $\underline{B}$ and DM be the multiplicative factor, drag coefficient translation, and Mach number translation, respectively, that when applied to the reference drag curve will

approximate the particular weapon drag curve of interest. That is:

$$Cd(M) \pm \underline{A} * \underline{Cd}(M + DM) + \underline{B}$$

A hypothetical weapon drag curve, Cd, and reference drag curve, Cd, are shown in Figure 3. For simplicity of discussion, the Mach number intervals (M2 -M1) and (M4 - M3) are equal. Suppose the Cd curve is multiplied by an A so that the Cd and the A * Cd curves are nearly identical. For the Cd and A * Cd curves to be a congruent, the A * Cd curve must be translated along the Mach axis an amount equal to DM, and along the Drag Coefficient axis an amount equal to B.

For many weapons, the above stated equation will not give an accurate fit over a wide interval of Mach numbers; instead, A, B and DM should be chosen so that the Cd fit is best for the Mach number interval that corresponds to the delivery envelope for a particular aircraft.

Considering the weapons currently implemented on a typical attack aircraft, (A7-E), both the Mk 84 and the Garve reference drag curves are necessary to handle the free-fall weapons. Instead of extending these curves to handle guns and rockets that have a much larger Mach number range, a rocket-gun reference drag curve is used.

F. CURVE FITTING THE DRAG COEFFICIENTS

In the previous section, the reference curves needed and how they were used to approximate weapon drag curves for use

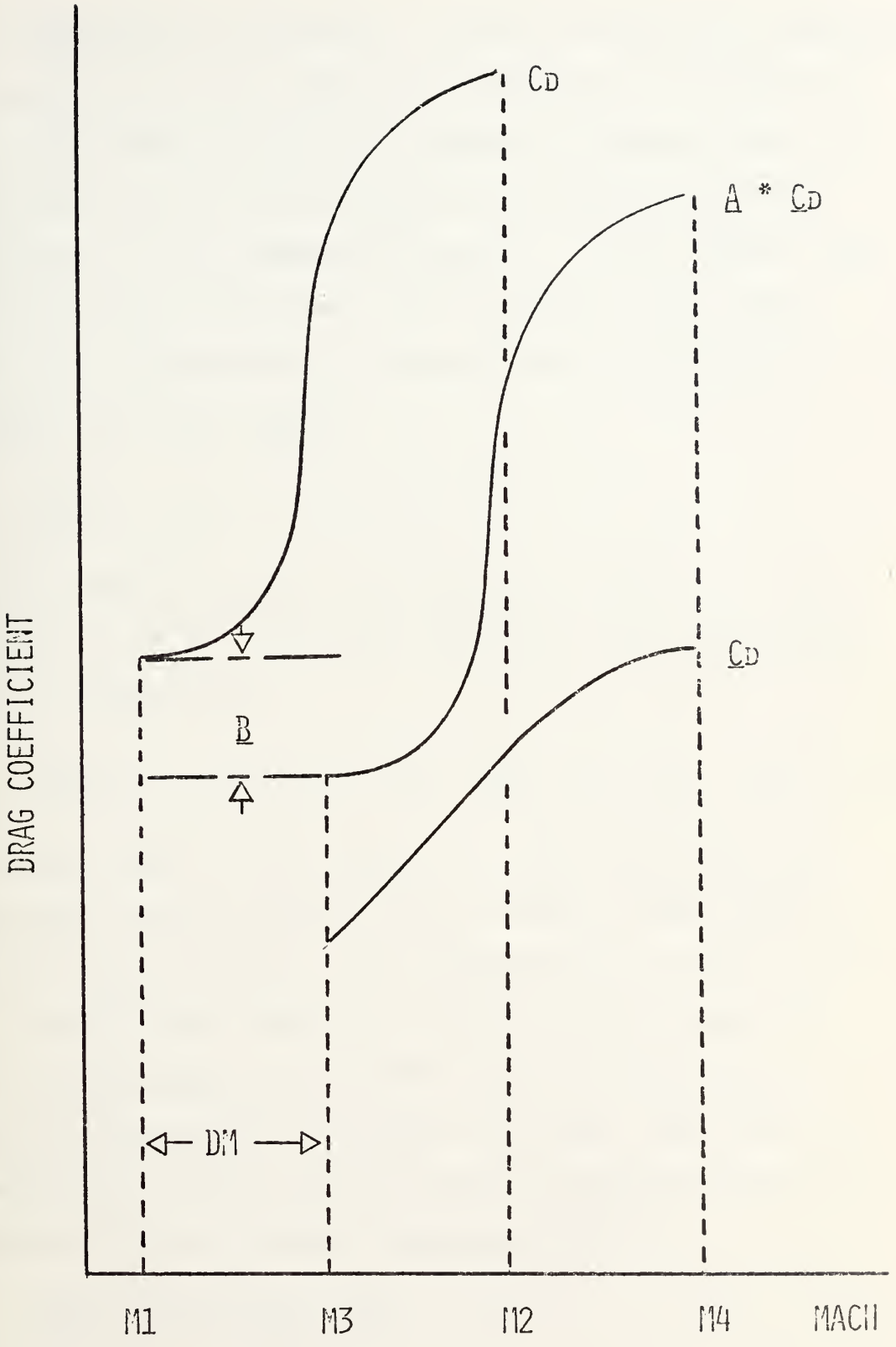


FIGURE 3. CURVE FITTING CD

in the algorithm was presented. In this section the reference drag curves versus Mach number is represented. The form of the function used will minimize computer storage, logic and computation time for the allowable error in downrange and time of fall for the different weapons. It was decided that each of the three reference curves be divided into three regions with Cd expressed as a second-order polynomial in Mach number for each region.

$$\underline{Cd}(M) = a0 + a1 * M + a2 * M^2$$

The coefficients, a0, a1, and a2 depend on the region the Mach number M is in. The drag coefficient for a particular weapon is expressed as:

$$Cd(M) \doteq \underline{A} * (a0 + a1 * (M + DM) + a2 * (M + DM)^2) + \underline{B}$$

As a further step to save computer storage and reduce the computation time for the algorithm, another compression of drag data was undertaken. The computation C * (PI/8 * Cd), CKDG, is made in the Runge-Kutta portion of the algorithm where C is the bomb factor, d^2/W, where d is the weapon diameter in feet and W is the mass in slugs. One constant per weapon in computer storage and two multiplications per integration step is saved by expressing CKDG as

$$CKDG = CF * \underline{CKDG} + DKG$$

where $CF = \underline{A} * (d/\underline{d})^2 * \underline{W}/W$

$$\underline{CKDG} = \underline{d}^2/\underline{W} * PI/8 * \underline{Cd}$$

$$DKG = d^2/W * PI/8 * \underline{B}$$

In the above equations, $\underline{d}$ and $\underline{W}$ are the assumed diameter and mass for the reference curve $\underline{Cd}$. The final form of CKDG in the algorithm is

$$CKDG(M) = CF * (b0 + b1 * (M + DM) + b2 * (M + DM)^2) + DKG$$

Table 1 gives a list of b0, b1, and b2 for each region of the three reference curves and the values of CF, DM, and DKG, are tabulated in Table 2.

Thrust appears in the algorithm in the total drag function

$$HH = TH/V - RHO * CKDG * V$$

where TH = thrust/mass

RHO = air density

V = weapon velocity

and CKDG was previously defined in the above equations. Figure 4 shows thrust versus time for a typical rocket approximated by a constant, thrust = total impulse/T1, where T1 is decreased from actual motor burn time in order to make thrust equal to the average thrust of region 2. The mass of the rocket, W, obviously decreases during rocket motor burn time. However, it is adequate for W to be

TABLE 1. Reference Drag Curve Coefficients and Cuts.

Coefficient	Reference Curve		
	Mk 84	Garve	Rocket-Gun
Region 1:			
b0	1.5729E-03	3.535039	0.104115
b1	0.0	-3.347782	-0.230347
b2	0.0	2.872624	0.167644
Region 2:			
b0	4.67840E-02	11.26165	-0.194037
b1	-0.109711	-27.41625	0.401478
b2	6.65480E-02	21.73083	-0.164612
Region 3:			
b0	-0.11638015	-23.79154	7.33246E-02
b1	0.21764389	44.26077	-2.03275E-02
b2	-9.76706E-02	-14.49960	2.44682E-03
Cut			
First cut, CT1	0.834	0.622	1.032
Second cut, CT2	0.977	0.885	1.300

TABLE 2. Weapon Constants in DECODE

IDNO	WEAPON	CFORM1	CFORM2	DKG1	DKG2	DM1	DM2	IREF	DS	DMAX	DTI	ITYPE
1	MK 43	0.0	0.0	2.55E-3	0.0	0.0	0.0	4	0.0	5.0	3.0	-1
2	MK 57	0.0	0.0	6.29E-3	0.0	0.0	0.0	4	0.0	5.0	3.0	-1
3	MK 61	0.0	0.0	4.01E-3	0.0	0.0	0.0	4	0.0	5.0	3.0	-1
4	MK 116	3.923E-3	0.0	2.75E-3	0.0	0.0	0.0	2	0.0	3.0	2.0	-1
5	MK 76	3.907E-3	0.0	6.36E-3	0.0	0.0	0.0	2	0.0	5.0	3.0	-1
6	MK 77	0.0	0.0	0.02126	0.0	0.0	0.0	4	0.0	2.0	1.0	-1
7	MK 81	2.5704	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
8	MK 81 SEF	0.0	0.0	9.76E-3	0.0	0.0	0.0	4	0.0	3.0	2.0	-1
9	MK 82 MEC	2.064	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
10	MK 82 ELE	1.4932	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
11	MK 83 MEC	1.3431	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
12	MK 83 ELE	1.21	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
13	MK 84	1.0	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
14	MK 117	3.12	0.0	-1.22E-3	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
15	MK 86	3.4972	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
16	MK 88	1.605	0.0	0.0	0.0	0.0	0.0	1	0.0	5.0	3.0	-1
17	MK 82 SEF	0.0	0.0	7.32E-3	0.0	0.0	0.0	4	0.0	3.0	1.0	-1

TABLE 2. Weapon Constants in DECODE (continued)

IDNO	WEAPON	CFORM1	CFORM2	DKG1	DKG2	DM1	DM2	IREF	DS	DMAX	DTI	ITYPE
18	MK 82 SERE	0.0	1.689E-2	7.32E-3	0.17166	0.0	0.32	1,2	0.66	0.0	2.0	1
19	SADEYE	2.0754	0.2217	0.0	0.0	0.0	0.0	1,2	4.27	0.0	1.5	1
20	ROCKEYE II	2.2973	1.114E-2	8.17E-3	0.16885	0.32	0.41	1,2	4.06	0.0	2.0	1
21	CBU	2.2404	0.1178	0.0	0.0	0.0	0.0	1,2	4.0	0.0	1.6	1
22	MK 81 SERE	0.0	2.306E-2	9.76E-3	0.23287	0.0	0.38	1,2	0.68	0.0	1.6	1
23	GUN	2.9964	0.0	-0.01499	0.0	0.0	0.0	3	0.0	1.5	0.5	-1
24	ROCKETS	0.82	1.0	0.0	0.0	0.0	0.0	3	1.42	0.0	1.0	2
25	MK 43 RET	0.0	0.0	0.0	1.48	0.0	0.0	4	0.98	0.0	0.3	0
26	MK 57 RET	0.0	0.0	0.0	2.0	0.0	0.0	4	0.89	0.0	0.2	0
27	MK 61 RET	0.0	0.0	0.0	2.7	0.0	0.0	4	0.89	0.0	0.1	0
28	MK 106	0.1514	0.1514	0.0	0.0	0.0	0.0	2	0.50	0.0	0.8	2

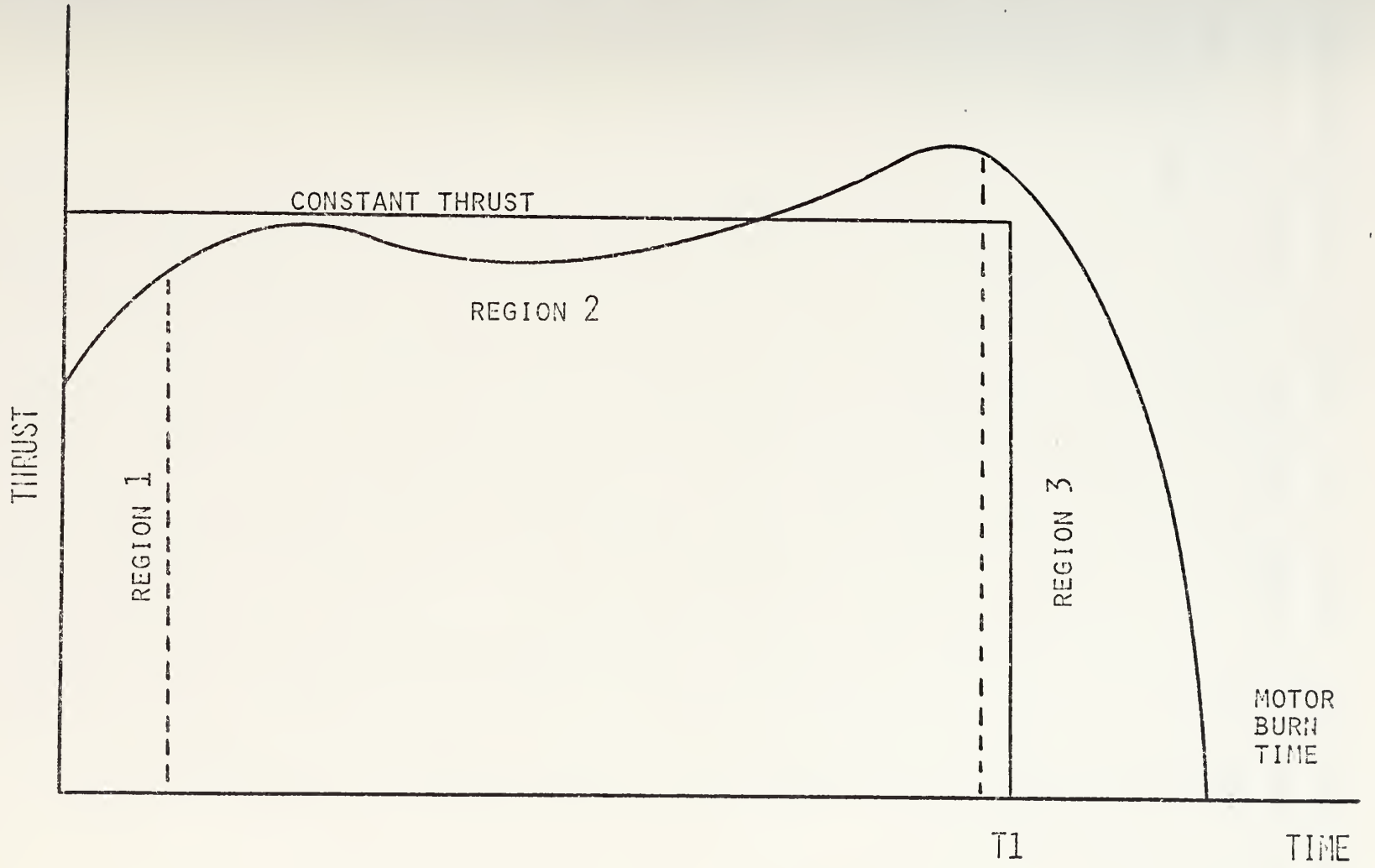


FIGURE 4. THRUST VERSUS TIME CURVE FOR A TYPICAL ROCKET

approximated by a constant, by varying W until downrange error is minimized. The second stage of the rocket, for time after weapon release greater than T1, is free fall and TH will be zero.

IV. FORTRAN SIMULATION

A. BOMB DROP SIMULATION

The initial ballistic trajectory algorithm for digital airborne fire control computers used in the simulation of Traj, Deriv, and the Runge routines, was developed by the Boeing Company, Seattle, Washington. The research was completed under a Naval Weapons Center contract in September 1970.

As previously mentioned, the Second-Order Runge-Kutta integration scheme was chosen because of its flexibility and simplicity. The results, as evidenced in Tables 3, 4 and 5, compared favorably to the published data (NAVAIR 01-1C-1T-1), with mean errors of 0.10% for the down range travel. The low-drag bombs showed relatively little sensitivity to the step size, so long as the maximum integration step, (DMAX), was less than or equal to 5 seconds and the normal Initial Integration Step, (DTI), was less than or equal to 3 seconds. However, the dual stage weapons (IDNO>17) did show a large sensitivity to the step size used during the integration scheme. The accuracy desired, required the DTI value to be consistently less than or equal to 2 seconds with the average value of 1.05 seconds, for the final eleven weapons.

The ultimate verification of the step size for the Runge-Kutta (Deriv also) method was conducted by varying

TABLE 3. Sample Errors in Impact Range Calculations for the General-Purpose Bomb Mk 82/Snakeye/Unretarded.

Release Conditions			Impact Range, ft		Impact Range error, ft
Velocity, knots	Altitude, ft	Angle, deg	Ballistic Tables	Algorithm	
400	2500	0	8039	8039	0
	5000	-10	9158	9159	+1
	12000	-30	10536	10537	+1
	12000	-45	7530	7531	+1
450	2500	0	8998	8995	-3
	5000	-10	10025	10023	-2
	12000	-30	11340	11341	+1
	12000	-45	7998	7999	+1
500	2500	0	9946	9940	-6
	5000	-10	10843	10839	-4
	12000	-30	12067	12067	0
	12000	-45	8406	8407	+1

TABLE 4. Sample Errors in Impact Range Calculations for the Special-Purpose Bomb Mk 40/Mod 0/Conical Tail.

Release Conditions			Impact Range, ft		Impact Range error, ft
Velocity, knots	Altitude, ft	Angle, deg	Ballistic Tables	Algorithm	
400	2500	0	8305	8301	-4
	5000	-10	9434	9432	-2
	12000	-30	10847	10846	-1
	12000	-45	7691	7690	-1
450	2500	0	9329	9324	-5
	5000	-10	10342	10339	-3
	12000	-30	11670	11669	-1
	12000	-45	8160	8159	-1
500	2500	0	10348	10342	-6
	5000	-10	11200	11197	-3
	12000	-30	12401	12402	+1
	12000	-45	8559	8559	0

TABLE 5. Sample Errors in Impact Range Calculations for the Practise-Purpose Bomb Mk 76/ with Lug.

Release Conditions			Impact Range, ft		Impact Range error ft
Velocity, knots	Altitude, ft	Angle, deg	Ballistic Tables	Algorithm	
400	2500	0	7633	7617	-16
	5000	-10	8719	8710	-9
	12000	-30	9991	9992	+1
	12000	-45	7232	7235	+3
450	2500	0	8463	8459	-4
	5000	-10	9493	9487	-6
	12000	-30	10712	10720	+8
	12000	-45	7669	7674	+5
500	2500	0	9268	9255	-13
	5000	-10	10195	10192	-3
	12000	-30	11337	11340	+3
	12000	-45	8038	8039	+1

the time interval from one to five seconds. The TRAJ integration scheme was simultaneously tested using several release conditions to demonstrate the model's flexibility in simulating bomb loft, level flight release and dive bombing conditions. The emphasis, at this point, was on the number of functional calls to RUNGE for the projectile to reach the target altitude. The results, by altering Δt, showed changes ranging from 0.00% to 0.44% for the time of fall, and 0.00% to 0.46% for the downrange travel, with a substantial decrease in the number of calls to RUNGE also evidenced (Table 6).

B. OVERALL SIMULATION

The PRINT routine was a means of simulating the various cockpit displays in the tactical attack aircraft. The output consisted of the release altitude, dive angle, and airspeed, as well as the resulting downrange travel and the time of fall of the weapon.

The INPUT routine simulated the collecting and passing of the sensor data to the various processors. The sensors included in the attack aircraft are the Inertial Navigation System, Doppler Radar, Air Data and the Radar Set. Specifically, the velocity, dive angle, altitude and the identification number of the weapon (usually provided via the armament panel) are required for the execution of the program.

The SETDAT routine completed the interface between the executive computer and the ballistics computer. Its purpose was that of assigning the constants their specific nonvarying

TABLE 6. Fortran Test Cases (time interval, MK 83)

RELEASE CONDITIONS	BALLISTIC TABLES*	DT = 1.0	DT = 2.0	DT = 3.0	DT = 4.0	DT = 5.0
10° 350 kts 1000 ft	11.71 6701	11.71 0.00% (12) 6699.9 -.01%	11.70 -.08% (6) 6699.2 -.02%	11.70 -.08% (4) 6698.8 -.03%	11.69 -.17% (3) 6699.2 -.02%	11.69 -.17% (3) 6692.1 -.13%
10° 350 kts 3000 ft	17.29 9826	17.29 0.00% (18) 9824.5 -.01%	17.29 0.00% (9) 9823.8 -.02%	17.28 -.05% (6) 9822.9 -.03%	17.28 -.05% (5) 9820.2 -.05%	17.27 -.11% (4) 9818.4 -.07%
10° 650 kts 500 ft	13.89 14192.	13.88 -.07% (14) 14187 -.03%	13.87 -.14% (7) 14179 -.08%	13.86 -.21% (5) 14164 -.19%	13.84 -.35% (4) 14140 -.36%	13.80 -.64% (3) 14126 -.46%
10° 650 kts 3000 ft	20.85 20859.	20.84 -.04% (21) 20851. -.03%	20.83 -.09% (11) 20841 -.08%	20.80 -.23% (7) 20833 -.12%	20.81 -.19% (6) 20810 -.23%	20.79 -.28% (5) 20785 -.35%
0° 300 kts 3500 ft	14.85 7382	14.84 -.06% (15) 7380.8 -.01%	14.84 -.06% (8) 7380.4 -.01%	14.83 -.13% (5) 7380.7 -.01%	14.83 -.13% (4) 7379.6 -.03%	14.81 -.26% (3) 7381.9 0.00%
0° 300 kts 15000 ft	30.98 15164.	30.96 -.06% (31) 15157 -.04%	30.95 -.09% (16) 15158 -.04%	30.94 -.12% (11) 15158 -.03%	30.92 -.19% (8) 15159 -.03%	30.93 -.16% (7) 15158 -.03%
0° 450 kts 5000 ft	17.82 13127.	17.82 0.00% (18) 13125 -.01%	17.81 -.05% (9) 13125 -.01%	17.80 -.11% (6) 13125 -.01%	17.80 -.11% (5) 13121 -.04%	17.79 -.16% (4) 13120 -.05%
-20° 400 kts 5500 ft	12.81 7963	12.81 0.00% (13) 7961.8 -.01%	12.81 0.00% (7) 7961.5 -.01%	12.81 0.00% (5) 7960.9 -.02%	12.80 -.07% (4) 7959.8 -.04%	12.79 -.15% (3) 7960.3 -.03%
-20° 550 kts 7000 ft	13.50 11387.	13.49 -.07% (14) 11386 -.01%	13.48 -.14% (7) 11386 -.01%	13.48 -.14% (5) 11385 -.01%	13.47 -.22% (4) 11384 -.02%	13.44 -.44% (3) 11387 0.00%

TABLE 6. Fortran Test Cases (continued)

RELEASE CONDITIONS	BALLISTIC TABLES*	DT = 1.0	DT = 2.0	DT = 3.0	DT = 4.0	DT = 5.0
-40° 450 kts 5000 ft	8.18 4689	8.18 0.00% (9) 4689.1 0.0%	8.18 0.00% (5) 4688.8 0.00%	8.17 -.12% (3) 4689. 0.00%	8.17 -.12% (3) 4687.7 -.02%	8.16 -.12% (2) 4689.1 0.0%
-60° 400 kts 7000 ft	9.61 3194	9.61 0.00% (10) 3193.4 -.01%	9.60 -.10% (5) 3193.4 -.01%	9.60 -.10% (4) 3193.2 -.02%	9.60 -0.10% (3) 3193.1 -.02%	9.57 -.41% (2) 3194.7 +.02%

* Navair 01-1C-1T-1

SAMPLE ENTRY

Time of fall	Test DT-Pub DT / Pub DT
(No. of calls to Runge Integration)	
Down Range Travel	Test DR-Pub DR / Pub DR

values and reinitializing the variables to zero at the start of each new weapon's integration. The routine also performed various calculations including the determination of the velocity in the x and y directions, the conversion of raw velocity to feet per second and the calculation of the release angle in radians.

The weapon dependent constants required for the algorithm were stored in DECODE. Also, DECODE contains the necessary logic to load the correct constants for the particular weapon the crew has selected. After the first pass, there is no need to go through DECODE before entering the algorithm for a trajectory calculation. The actual scheme used for DECODE in reality will depend on the particular aircraft and its fire control requirements and how the algorithm is interfaced in the airborne computer.

V. PL/M IMPLEMENTATION

To implement the model on the INTEL 8080 microcomputer, the floating point arithmetic package was used to maintain a four hexadecimal digit accuracy during calculations. The floating point variable therefore occupies three eight-bit words: two words for the variable mantissa and one word for the mantissa sign bit and the variable exponent. The mantissa of the variable is viewed as a normalized binary number with the left most digit always one.

In order to manipulate these variables, an arithmetic subroutine library was supplied by Professor Kodres. At present, there exist routines that add, subtract, multiply, divide and take the square roots of these floating point variables.

The actual PL/M version of the ballistics program was translated directly from the working FORTRAN version. All the associated flow charts, process graphs, program listings, and the simulated output can be found in the appendix.

The PL/M routines have been compiled and tested. The results compare very favorably with the FORTRAN results. The PL/M verification of the test parameters is included in Table 7.

TABLE 7. PL/M Test Verification (time interval, MK 83)

RELEASE CONDITIONS	BALLISTIC TABLES	DT = 3.0	COMPUTATION TIME	
			Trial 1	Trial 2
10° 350 kts 1000 ft	11.71 6701.	11.69 -.17% (4) 6694.5-.09%	1.2 (sec)	1.1 (sec)
10° 350 kts 3000 ft	17.29 9826.	17.28 -.05% (6) 9820.5-.05%	1.8	1.7
10° 650 kts 500 ft	13.89 14192.	13.82 -.49% (5) 14140.-.36%	1.3	1.2
10° 650 kts 3000 ft	20.85 20859.	20.81 -.19% (7) 20820.-.18%	2.2	2.1
0° 300 kts 3500 ft	14.85 7382.	14.83 -.13% (5) 7379. -.03%	1.6	1.4
0° 300 kts 15000 ft	30.98 15164.	30.95 -.09% (11) 15156.-.04%	3.2	3.0
0° 450 kts 5000 ft	17.82 13127.	17.81 -.05% (6) 13122.-.03%	1.8	1.7
-20° 400 kts 5500 ft	12.81 7963.	12.80 -.07% (5) 7963. 0.00%	1.2	1.1
-20° 550 kts 7000 ft	13.50 11387.	13.47 -.22% (5) 11388.+.01%	1.2	1.2
-40° 450 kts 5000 ft	8.18 4689.	8.18 0.00% (3) 4689. 0.00%	0.9	1.0
-60° 400 kts 7000 ft	9.61 3194.	9.60 -.10% (4) 3195. +.02%	0.8	0.8

VI. CONCLUSIONS

The objective of this thesis was to demonstrate a microcomputer based digital system which will perform the same functions as the attack portion of a presently implemented airborne tactical system. The main purpose of the project was to verify the feasibility of implementing the most complex arithmetic task of the airborne tactical computer by use of an INTEL 8080 microcomputer at an estimated cost of $3891 for the entire tactical system.

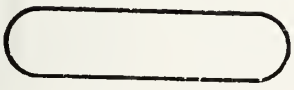
Because of the rapidly decreasing hardware costs of the microcomputer and significant increases in processing speeds which the microcomputer industry is presently realizing, these new, innovative systems, which are in all ways equivalent to the presently available avionics systems, can be developed at a hardware cost which is nearly 10 times smaller. The software development costs are also likely to be smaller because the system can be subdivided into smaller, independently operating subsystems, each of which will perform a well defined task.

The lack of hardware multiply-and-divide operations in the microcomputer is compensated by careful process analysis to reduce the number of multiply-and-divide operations. This allows three "slow" (in the computer sense) computers to solve the same problem that one "fast" computer is presently solving, at a great hardware cost reduction. The necessity for careful analysis of the problem may even disappear soon,

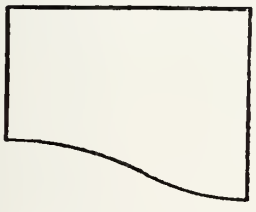
because the microcomputer manufacturers are likely to incorporate the hardware multiply-and-divide operations. This thesis demonstrates the feasibility of using the microcomputer as a systems' building block for airborne tactical systems.

APPENDIX A PROGRAM FLOW CHARTS AND PROCESS GRAPHS

Symbols Used In Flow Charts



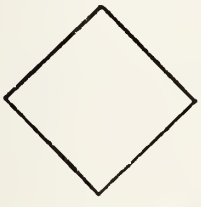
Start/Stop



Output

Card Input

Processing Symbol
(Arithmetic Operation)



Conditional Symbol

Subroutine Call

Return Statement

Program Entry Point (label)

Symbols Used in Process Graphs

1. Bite Operations (8 bits):

+	ADD
*	MULTIPLY
-	SUBTRACT
/	DIVIDE
=	ASSIGNMENT

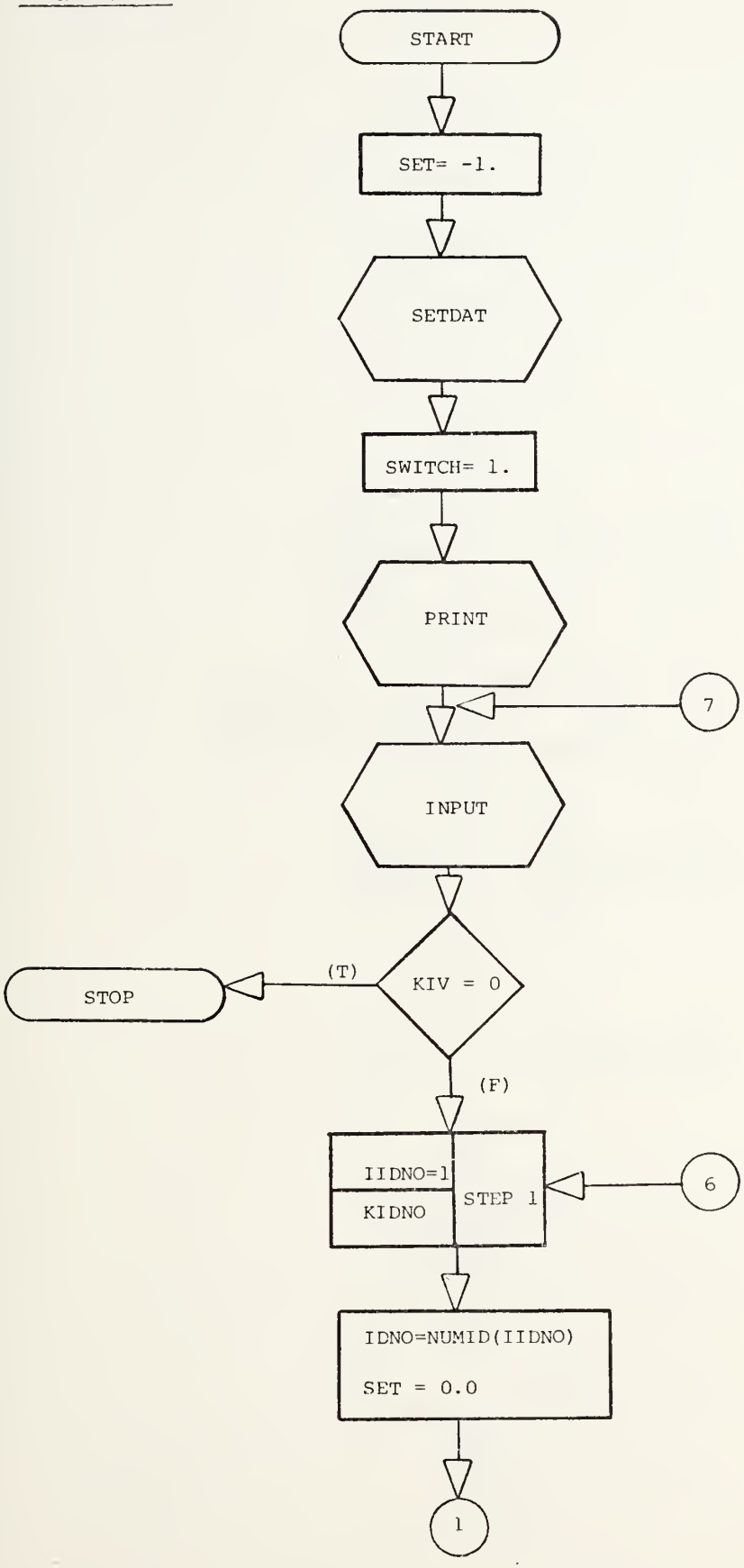
2. Floating Point Operations:

ADD	ADD (Subtract) Routine
MULT	MULTIPLY Routine
DIV	DIVIDE Routine
SQRT	SQUARE ROOT Routine

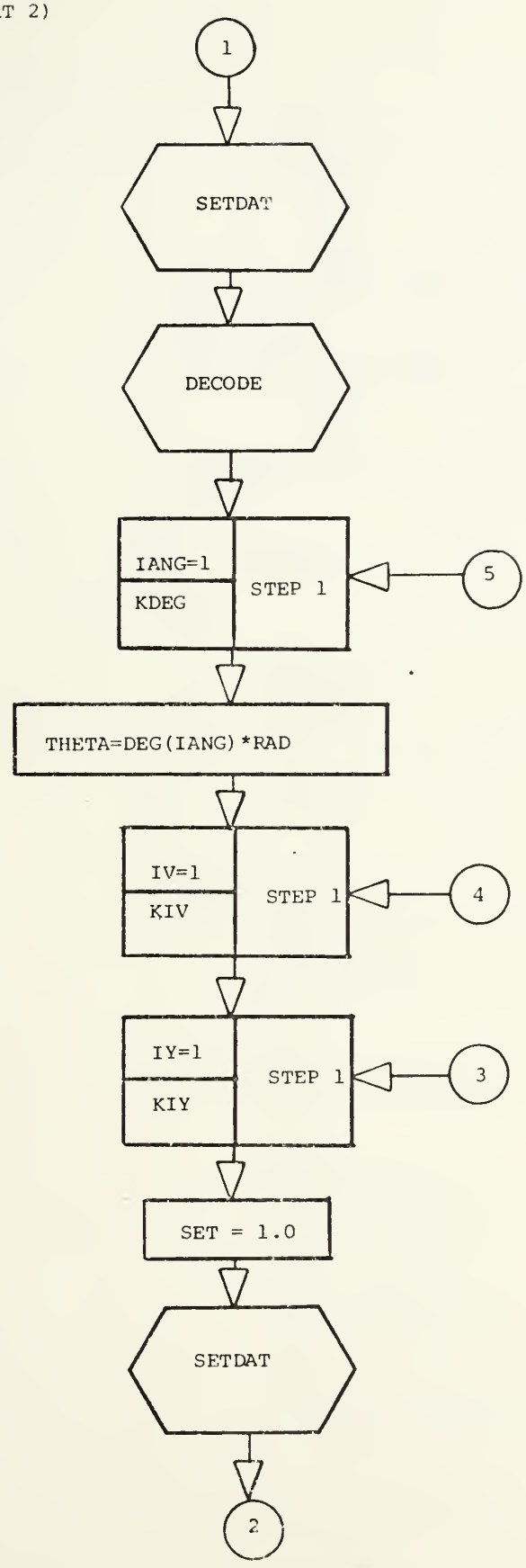
3. Subroutine Calls

DERIV	First Derivative Routine

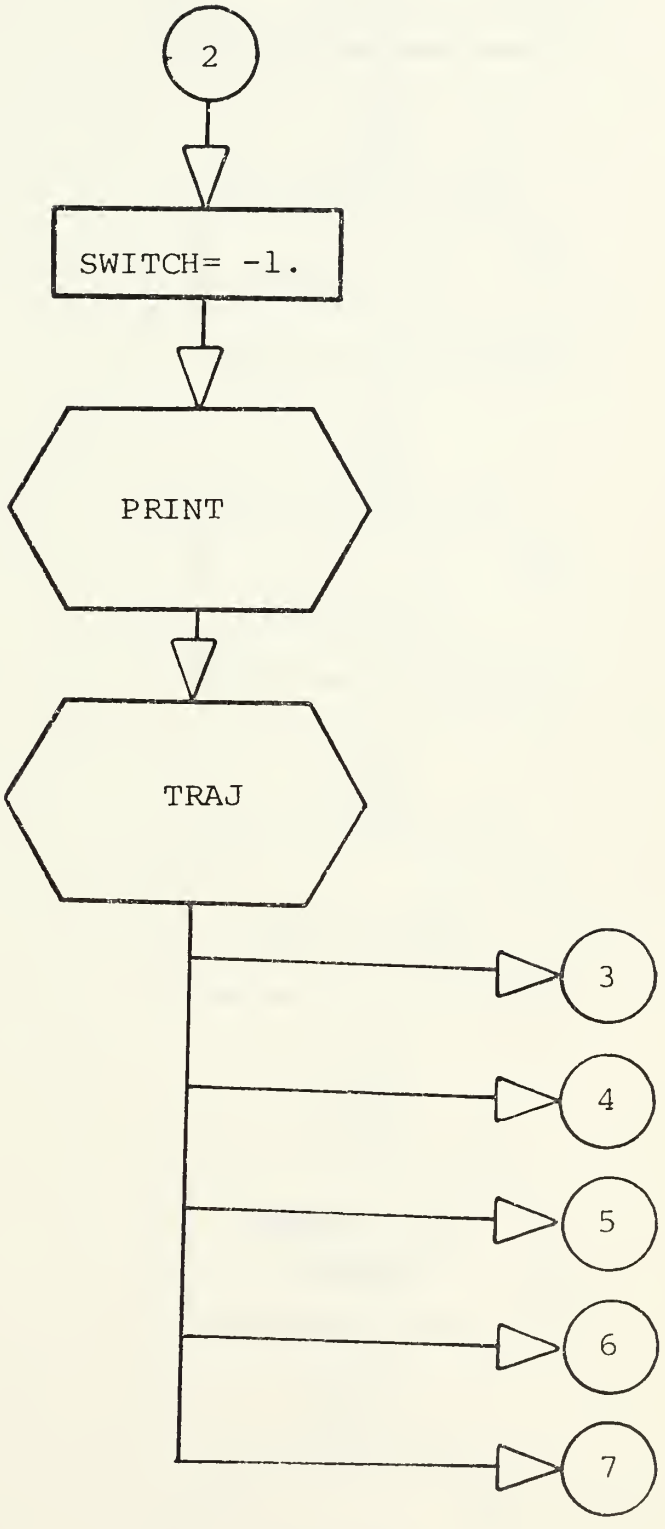
MAIN PROGRAM



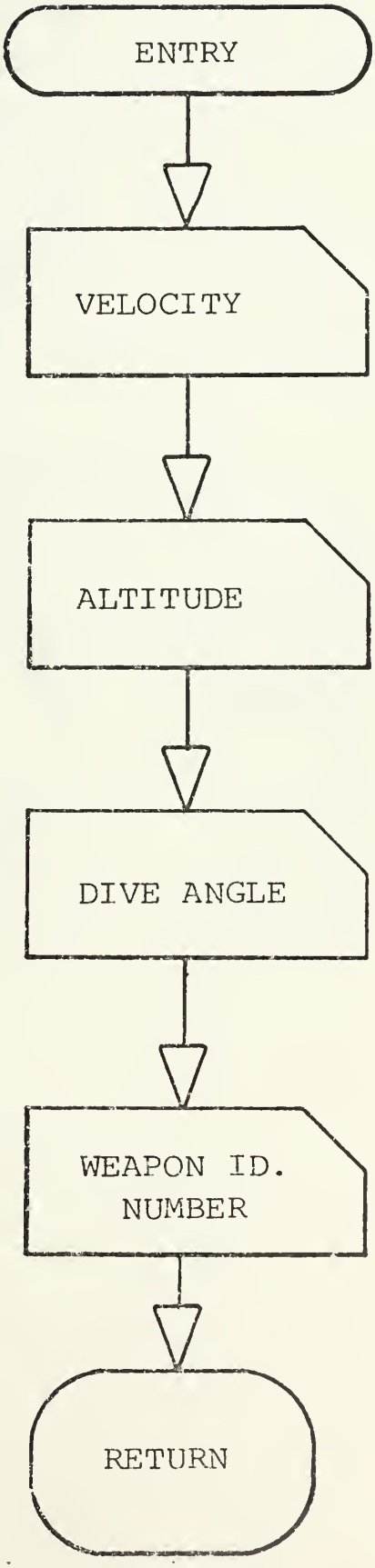
MAIN PROGRAM (PART 2)



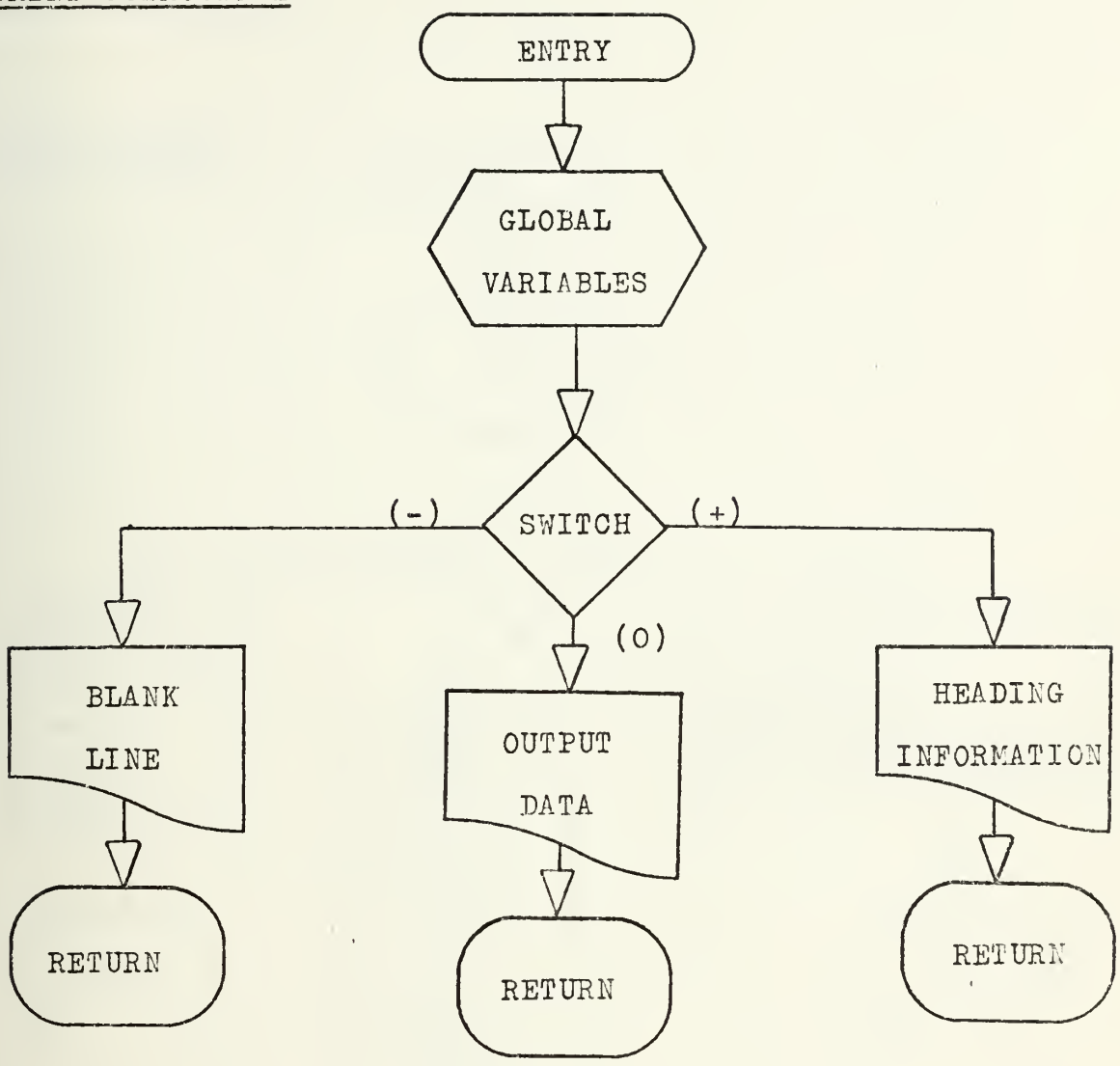
MAIN PROGRAM (PART 3)



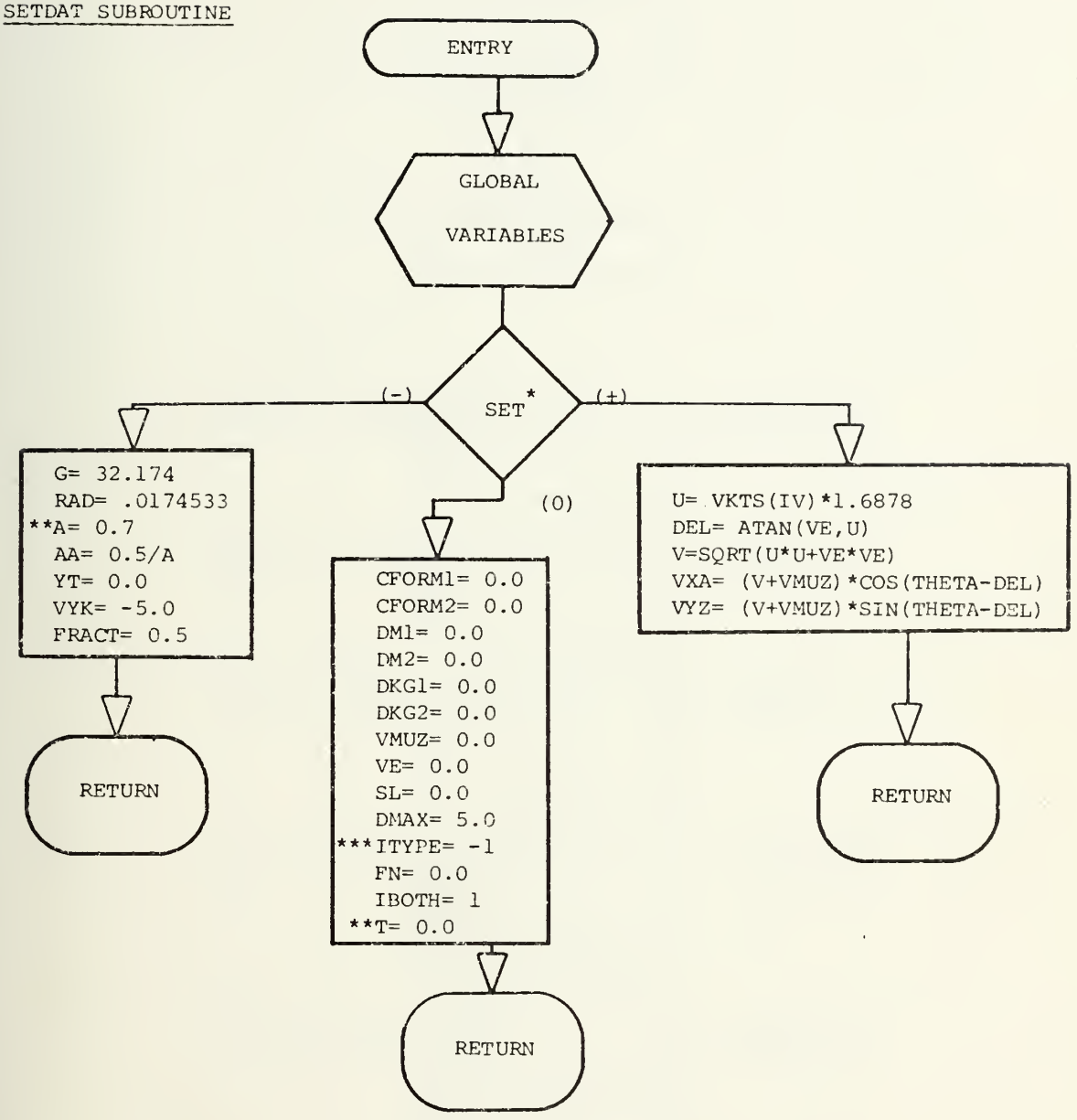
INPUT SUBROUTINE



PRINT SUBROUTINE



SETDAT SUBROUTINE

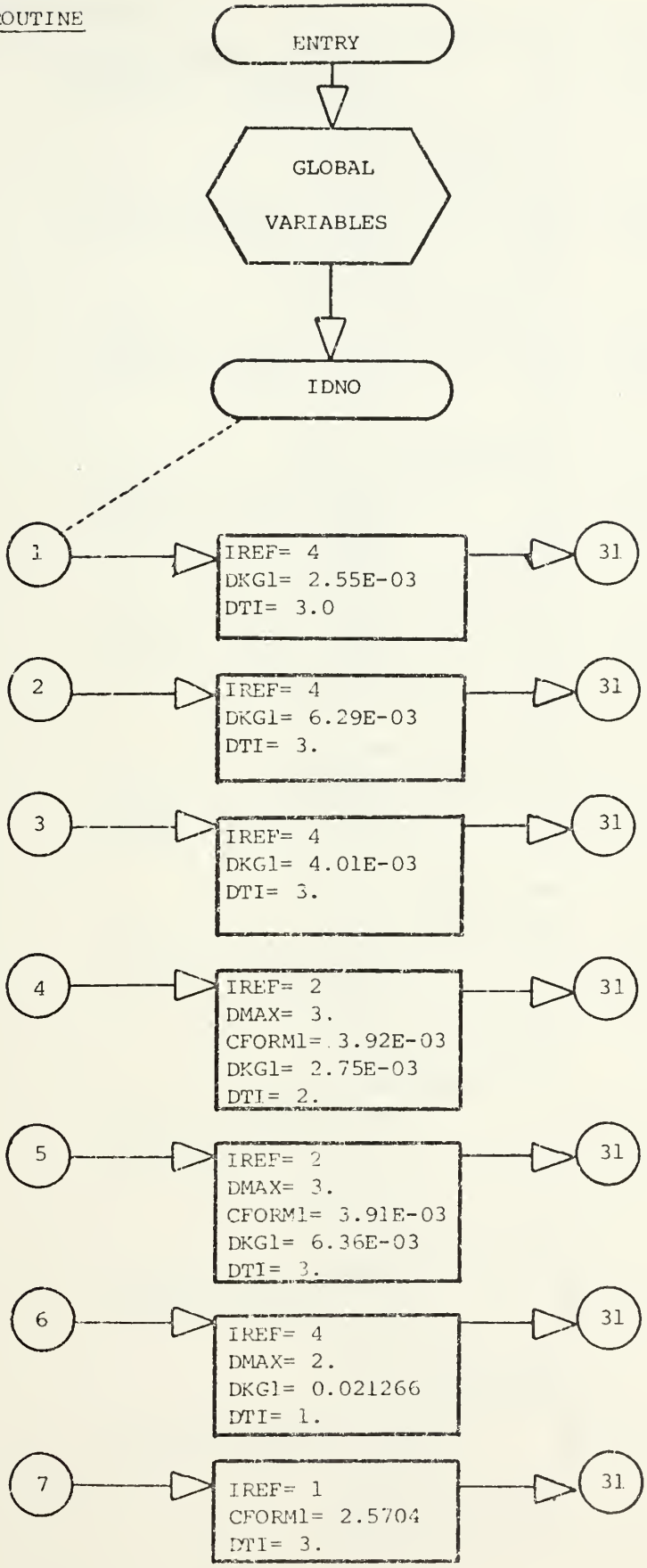


* SET VALUES FOR PL/M ARE 1, 2, 3

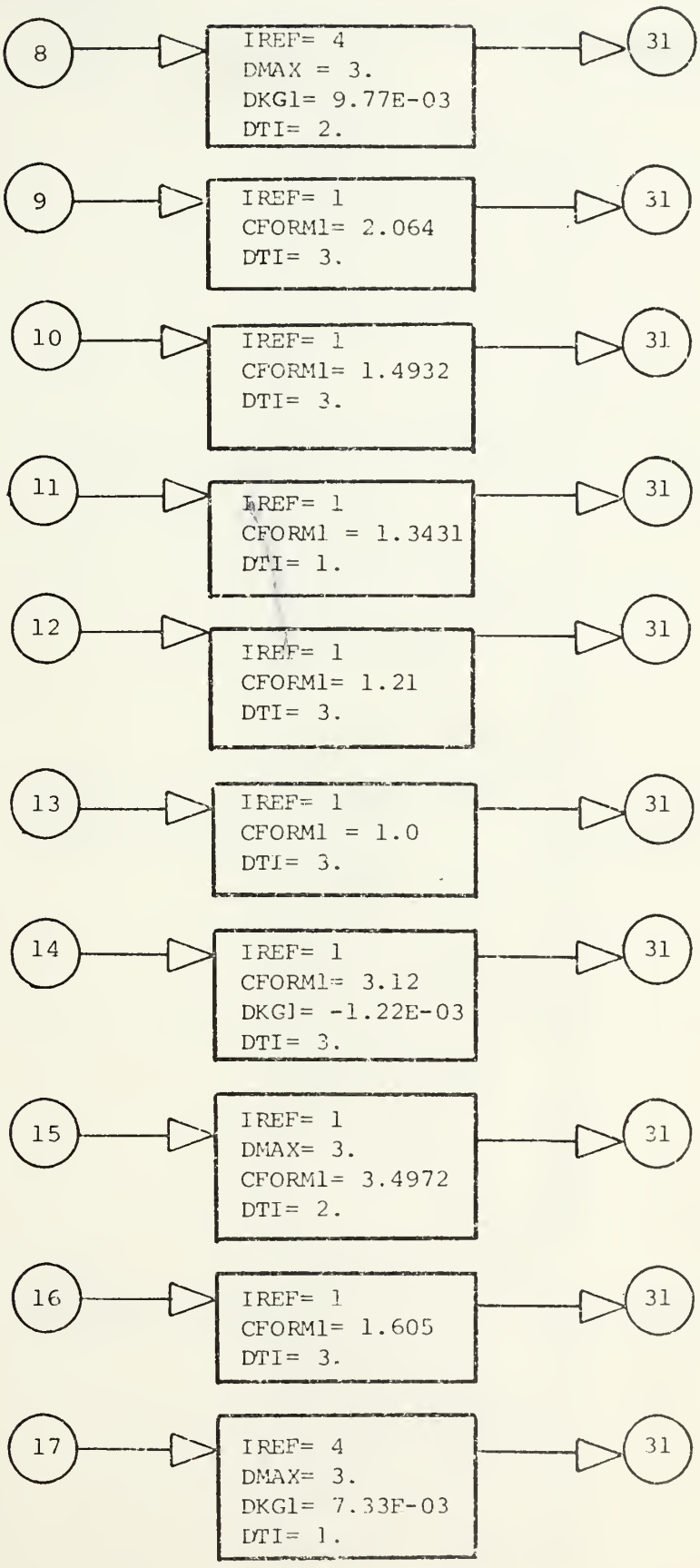
** T IS TM AND A IS A_1 IN PL/M.

*** ITYPE HAS A VALUE 3 FOR -1 IN PL/M.

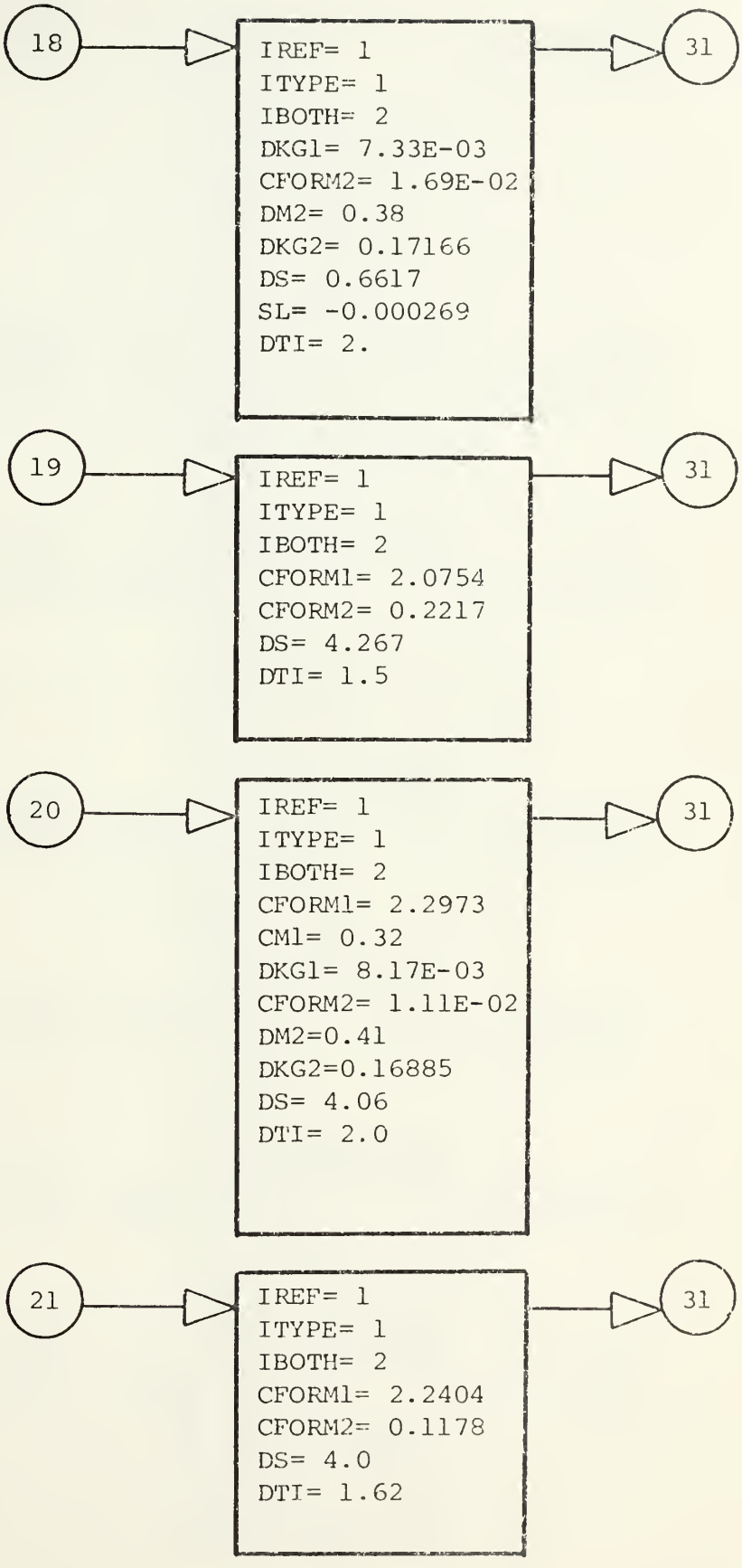
DECODE SUBROUTINE



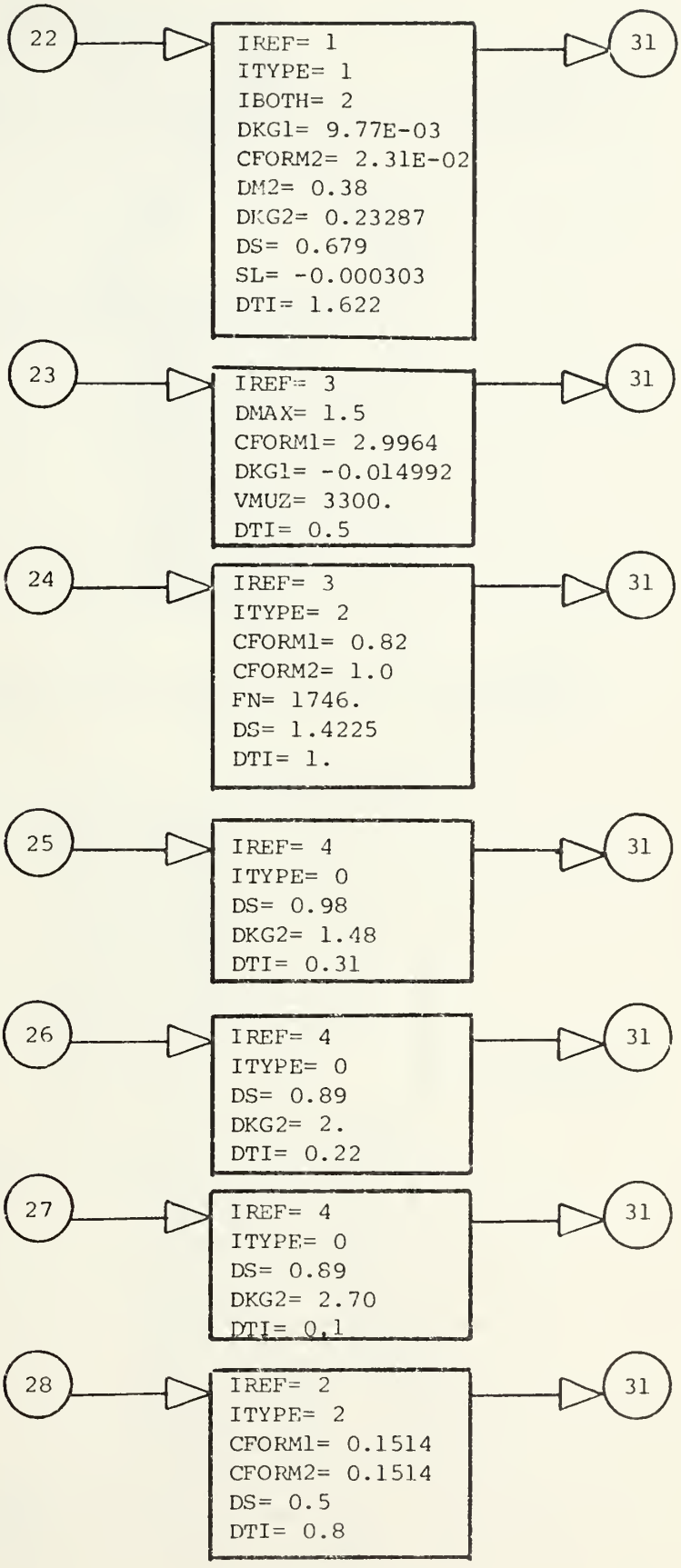
DECODE SUBROUTINE (PART 2)



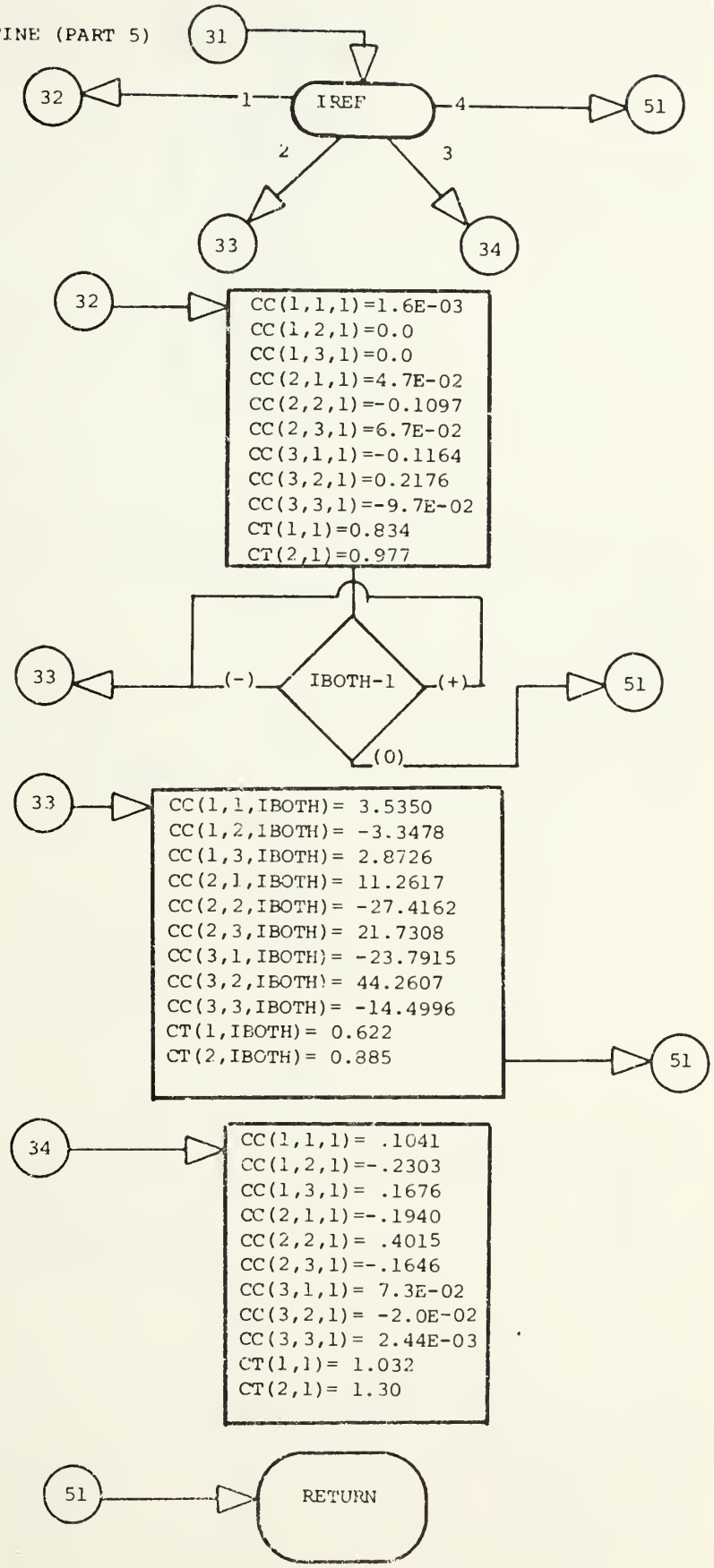
DECODE SUBROUTINE (PART 3)



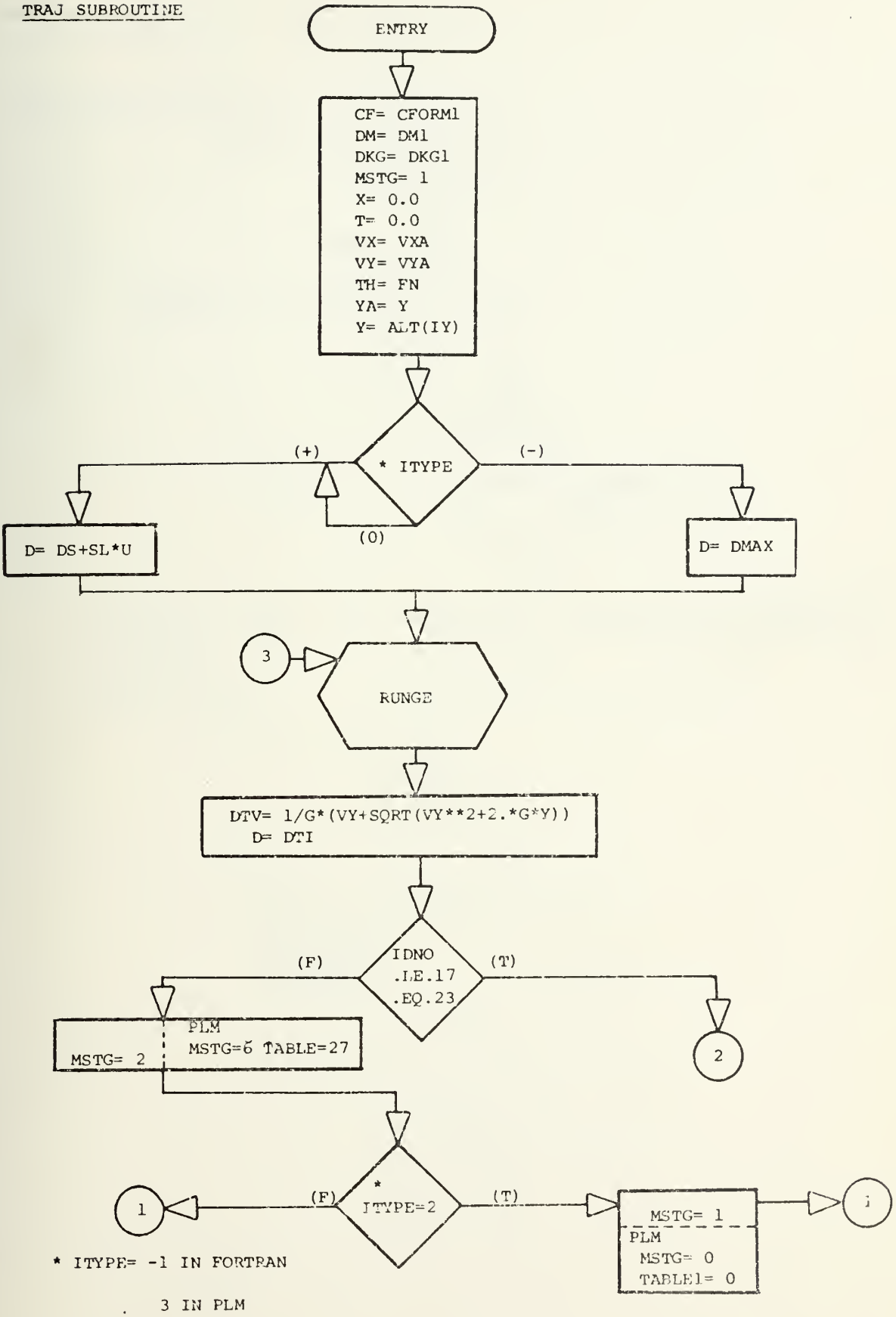
DECODE SUBROUTINE (PART 4)



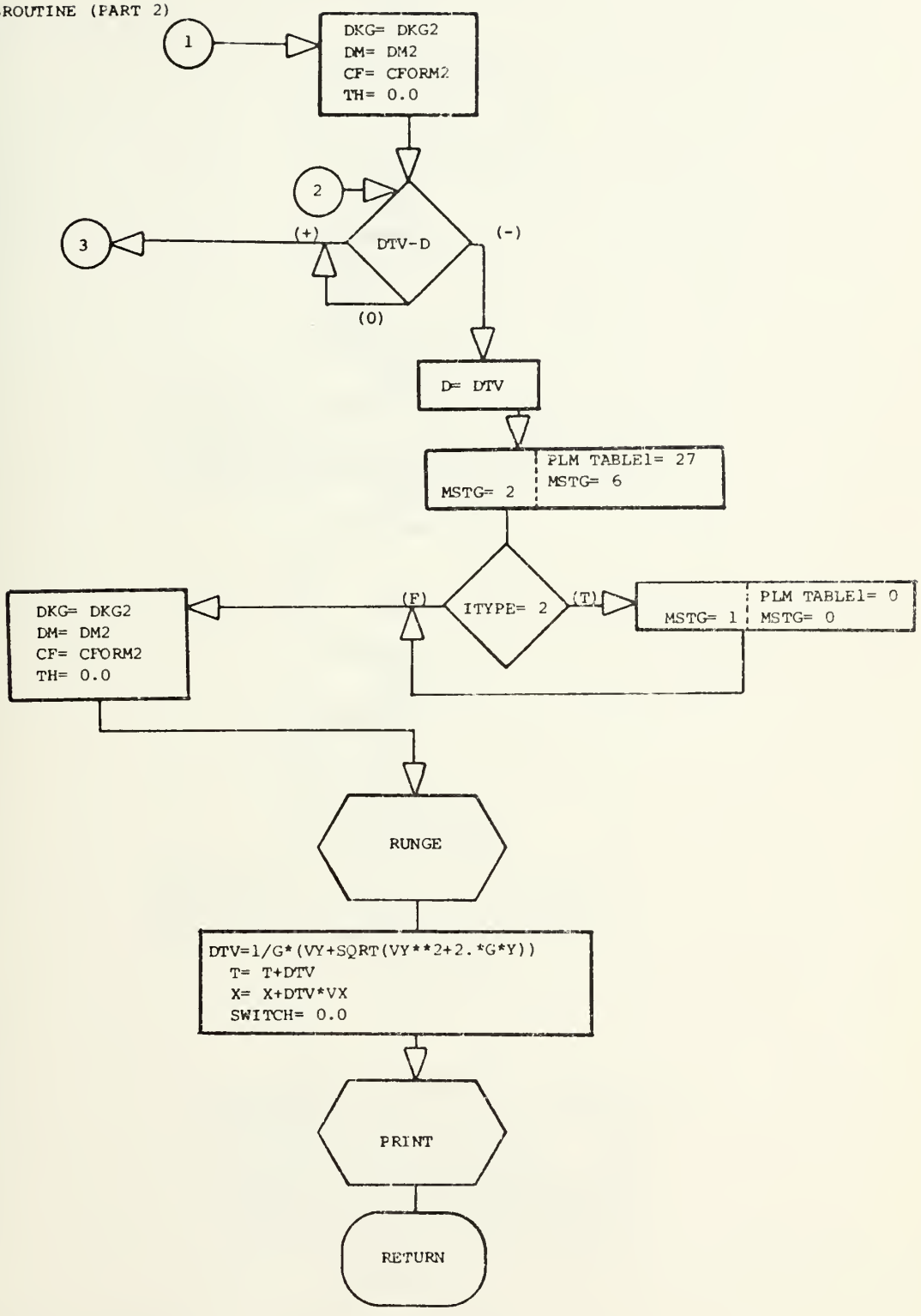
DECODE SUBROUTINE (PART 5)



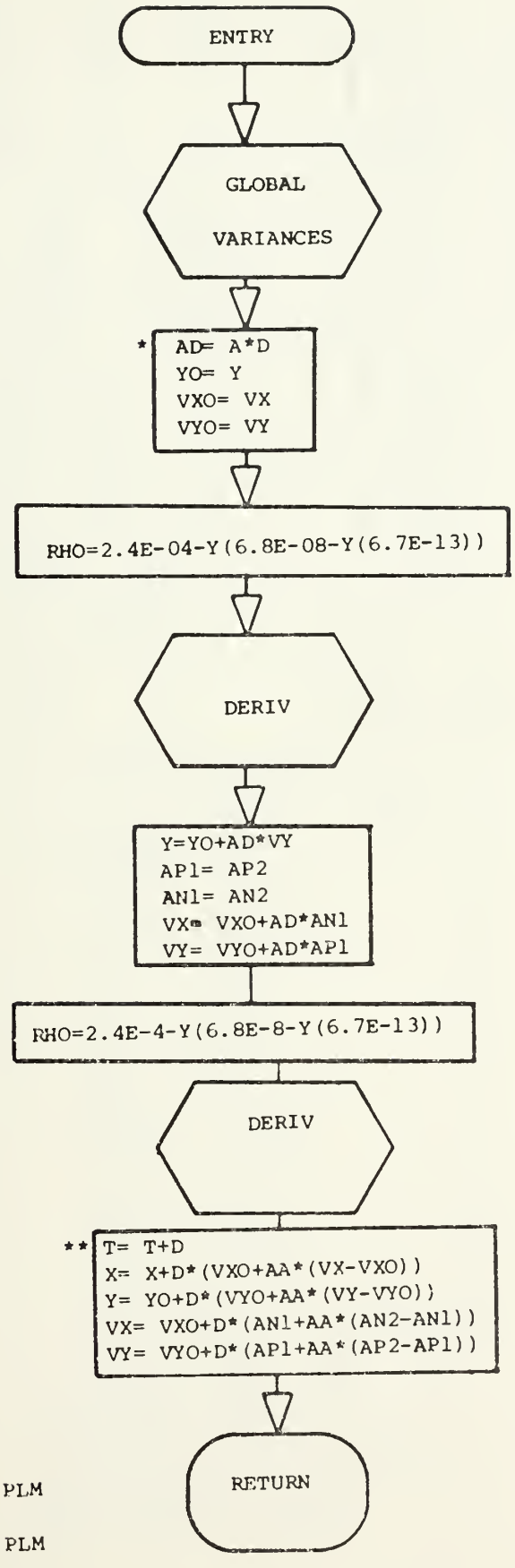
TRAJ SUBROUTINE



TRAJ SUBROUTINE (PART 2)



RUNGE SUBROUTINE

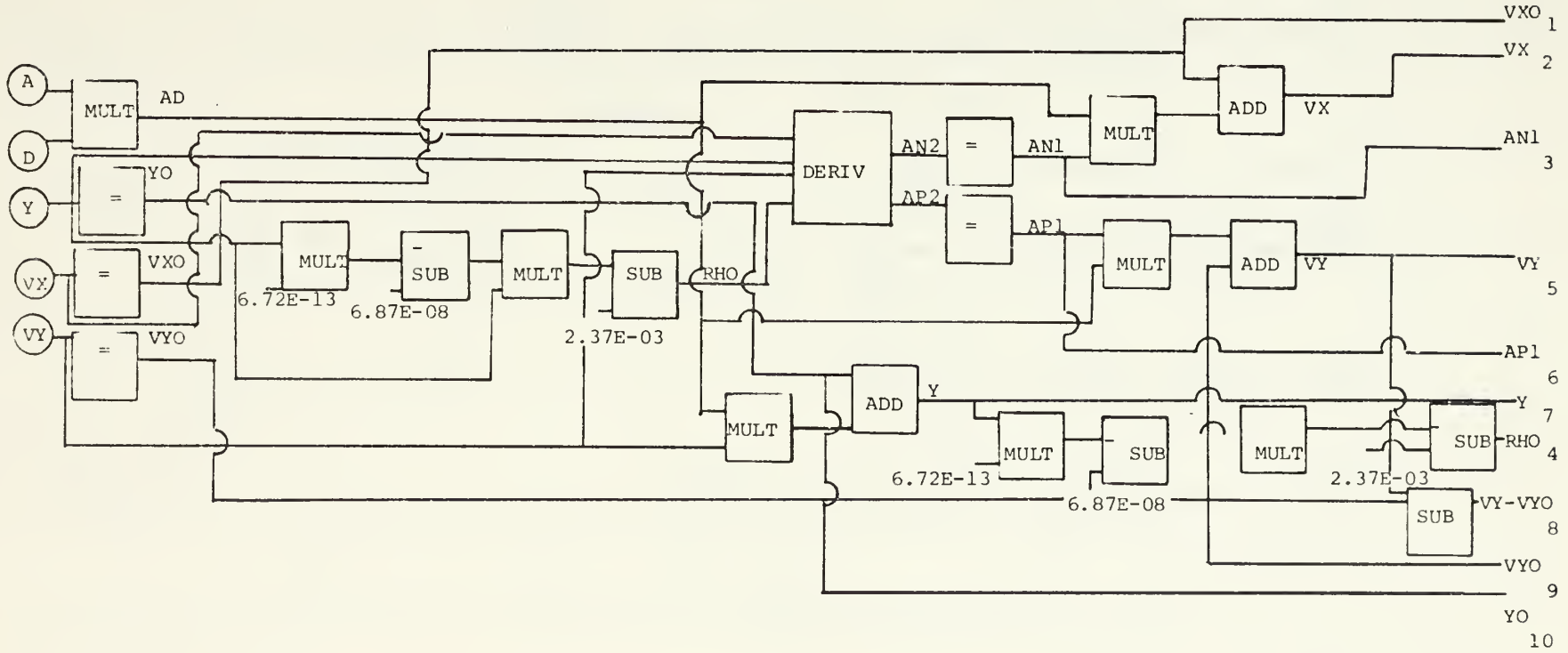


*A IS A_1 IN PLM

** T IS TM IN PLM

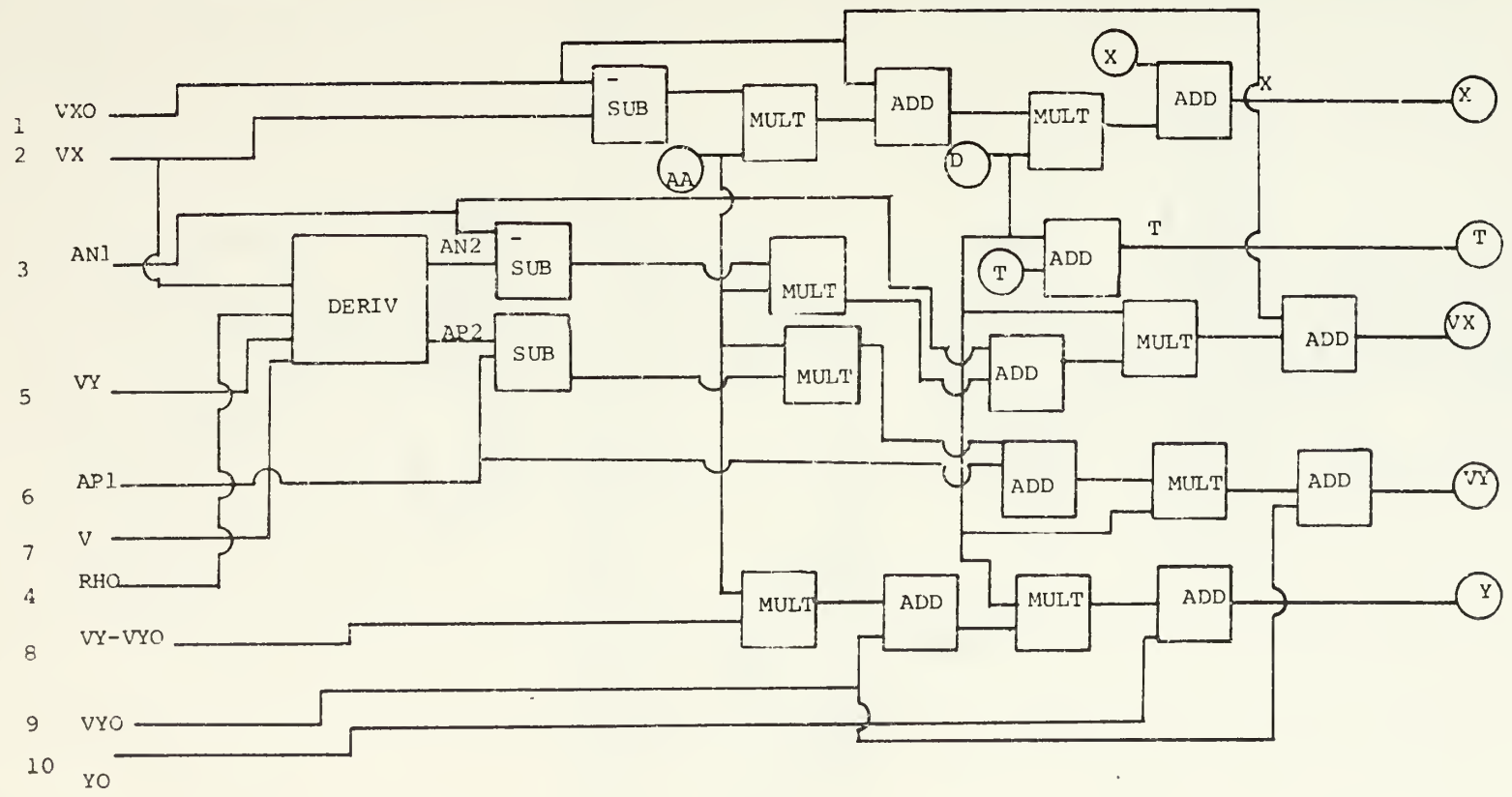
RUNGE PROCESS CHART

(PART 1)

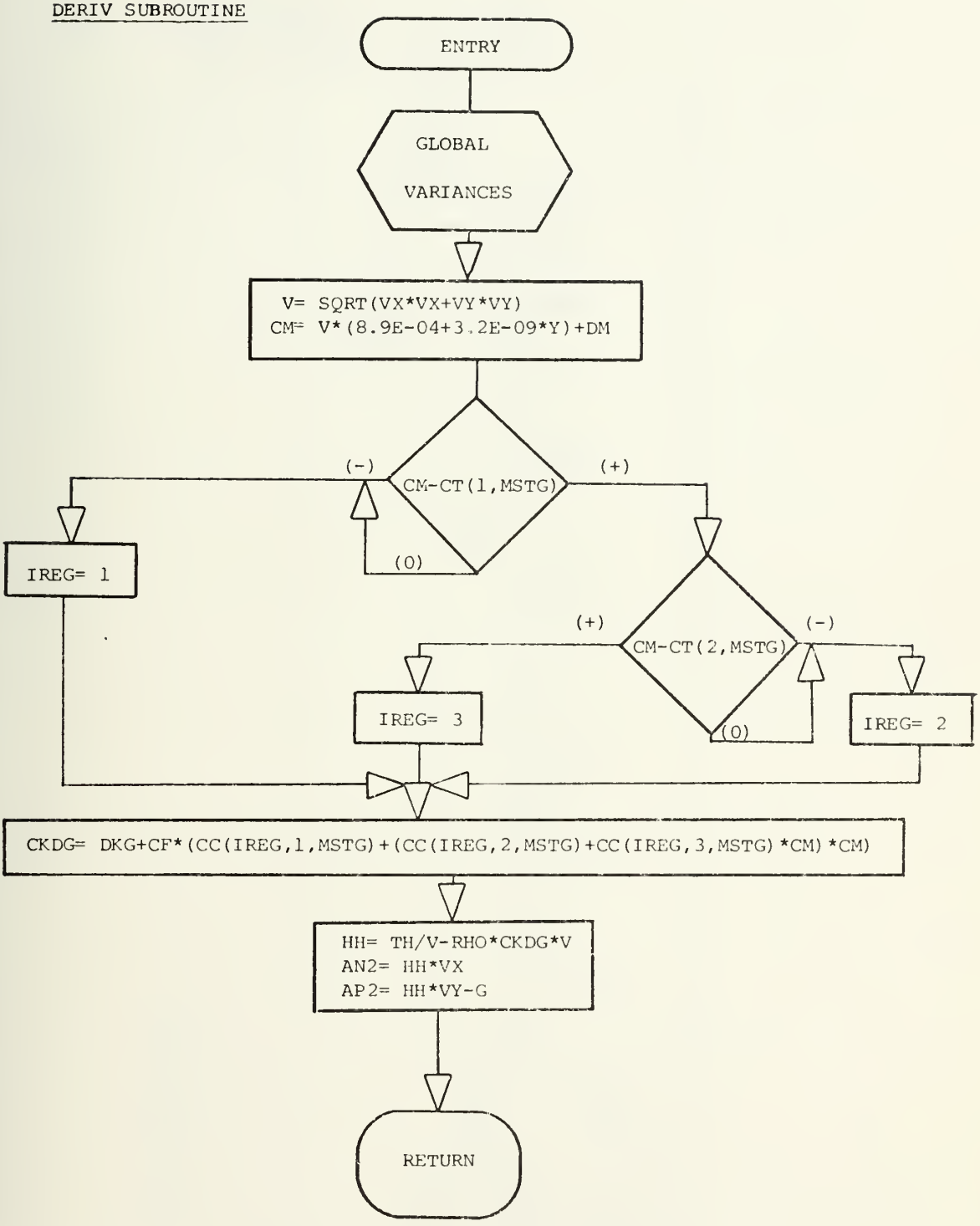


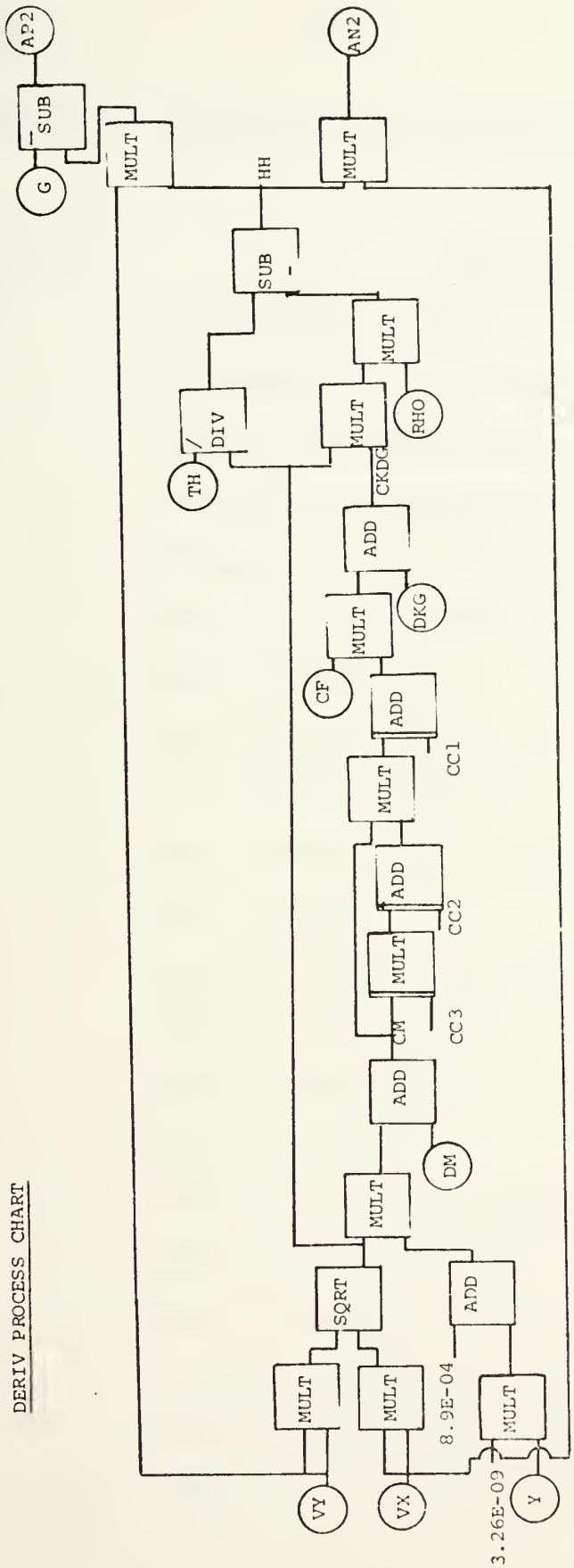
RUNGE PROCESS CHART

(PART 2)



DERIV SUBROUTINE




DERIV PROCESS CHART
AP2
SUB
G
MULT
HH
AN2
MULT
SUB
MULT
DIV
TH
MULT
RHO
CKDG
ADD
DKG
MULT
CF
ADD
CC1
MULT
ADD
CC2
MULT
CM
CC3
ADD
DM
MULT
SQRT
ADD
MULT
MULT
8.9E-04
MULT
3.26E-09
VY
VX
Y

APPENDIX B: PROGRAM DEFINITION OF VARIABLES

FORTRAN	SYMBOL PL/M if varied	DEFINITION
A	A1	Runge Kutta parameter
AA		0.5/Runge Kutta parameter
ALT		Weapon release altitude
AN1,AN2		Runge Kutta variables (x direction)
AP1,AP2		Runge Kutta variables (y direction)
CC		Matrix of drag curve coefficients
CF		CKDG stretch factor
CFORM1		CKDG stretch factor for the first stage
CFORM2		CKDG stretch factor for the second stage
CKDG		Bomb coefficient times drag coefficient
CM		Mach number
CT		Matrix of Mach curve coefficients
DEG		Weapon release angle
DEL		ArcTan (ejection velocity/weapon velocity)
DKG		Shift in CKDG
DKG1		Shift in CKDG for the first stage
DKG2		Shift in CKDG for the second stage
DM		Shift in Mach number
DMAX		Largest integration step size
DM1		Shift in first stage Mach number
DM2		Shift in second stage Mach number
DS		Integration step size

SYMBOL FORTRAN	SYMBOL PL/M if varied	DEFINITION
DTI		Initial integration step size
FN		Equal to the Thrust for the first stage
FRACT		Parameter for Runte Kutta integration (value between zero and one)
G		Acceleration due to gravity
HH		Total drag function
IANG		Number of release angles integrated
IBOTH		Key determining if two drag curves are required for the weapon
IDNO		Identification number of each weapon
IREF		Reference for Mach curve coefficients
IREG		Mach region index
IV		Number of velocities integrated
IY		Number of altitudes integrated
MSTG		Weapon stage index
RAD		Conversion factor for radians
RHO		Air density
SET		Key to initializing variables
SL		Slope coefficient for retarded snakeye fins deployment time
T	TM	Elapsed time of flight from weapon release
TH		Thrust/Mass of rocket
THETA		Adjusted weapon release angle
U		Aircraft speed
V		Velocity of weapon
VX		Velocity component of V in the x direction

FORTRAN	SYMBOL PL/M if varied	DEFINITION
VXA		VX at weapon release
VXO		VX at start of present integration step
VY		Velocity component of V in the y direction
VYA		VY at weapon release
VYO		VY at start of present integration step
VZ		Muzzle velocity of guns
X		Weapon ground range from release
Y		Weapon altitude above sea level
YO		Y at start of present integration step
YT		Target altitude above sea level

FORTRAN AND PL/M RESULTS

IDNO	VKTS	ALT	DEG	FORTRAN		PL/M	
				TOF	RANGE	TOF	RANGE
1	450.	3000.	-10.	10.26	7515.	10.27	7518.
2	450.	3000.	-10.	10.41	7393.	10.41	7395.
3	450.	3000.	-10.	10.32	7467.	10.32	7470.
4	450.	3000.	-10.	10.73	7189.	10.74	7191.
5	450.	3000.	-10.	10.91	7085.	10.87	7078.
6	450.	3000.	-10.	11.13	6955.	11.14	6957.
7	450.	3000.	-10.	10.32	7466.	10.33	7469.
8	450.	3000.	-10.	10.59	7292.	10.60	7294.
9	450.	3000.	-10.	10.29	7492.	10.29	7495.
10	450.	3000.	-10.	10.25	7521.	10.26	7524.
11	450.	3000.	-10.	10.25	7529.	10.26	7532.
12	450.	3000.	-10.	10.24	7536.	10.24	7539.
13	450.	3000.	-10.	10.22	7547.	10.23	7550.
14	450.	3000.	-10.	10.31	7478.	10.31	7480.
15	450.	3000.	-10.	10.40	7424.	10.41	7427.
16	450.	3000.	-10.	10.26	7516.	10.27	7519.
17	450.	3000.	-10.	10.50	7368.	10.51	7371.
18	450.	3000.	-10.	16.55	3852.	16.57	3847.
19	450.	3000.	-10.	20.84	4620.	20.87	4619.
20	450.	3000.	-10.	14.92	5771.	14.94	5768.
21	450.	3000.	-10.	16.83	5484.	16.87	5481.
22	450.	3000.	-10.	17.96	3299.	17.98	3296.
23	450.	3000.	-10.	10.67	10940.	10.69	10935.

FORTRAN AND PL/M RESULTS (continued)

IDNO	VKTS	ALT	DEG	FORTRAN		PL/M	
				TOF	RANGE	TOF	RANGE
24	450.	3000.	-10.	7.82	12814.	7.85	12815.
25	450.	3000.	-10.	31.70	1657.	31.70	1656.
26	450.	3000.	-10.	36.29	1384.	36.30	1382.
27	450.	3000.	-10.	41.58	1213.	41.58	1210.
28	450.	3000.	-10.	20.56	2444.	20.57	2442.

```
      COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),  JUP00010
     1IPEF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,         JUP00020
     2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,              JUP00030
     3RAD,  FRACT,  VE,  THETA, U, DEL, V, VXA, VYA,                     JUP00040
     4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,              JUP00050
     5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,       JUP00060
     6YO, VYO, AN1, AN2, AP1, AP2, IREG, RHO,DTI, DTV                    JUP00070
      SET = -1.0                                                         JUP00080
C CALL SETDAT TO INITIALIZE THE CONSTANTS                                JUP00090
      CALL SETDAT                                                        JUP00100
      SWITCH = 1.0                                                       JUP00110
C CALL PRINT TO PRINT THE HEADING INFORMATION                            JUP00120
      CALL PRINT                                                         JUP00130
  1CO CONTINUE                                                           JUP00140
C CALL INPUT TO UPDATE THE UNKNOWNS IN THE EQUATIONS                     JUP00150
      CALL INPUT                                                         JUP00160
C TEST ON KIV                                                            JUP00170
      IF (KIV.EQ.0) GO TO 9999                                           JUP00180
      DO 5 IIDNO=1, KIDNO                                                JUP00190
      IDNO= NUMID(IIDNO)                                                 JUP00200
      SET = 0.0                                                          JUP00210
C CALL SETDAT TO RE-INITIALIZE THE VARIABLES FOR EACH IDNO               JUP00220
      CALL SETDAT                                                        JUP00230
C CALL DECODE TO INITIALIZE THE BOMB COEFFICIENTS                        JUP00240
      CALL DECODE                                                        JUP00250
      DO 4 IANG=1,KDEG                                                   JUP00260
      DO 3 IV=1, KIV                                                     JUP00280
      DO 2 IY=1, KY                                                      JUP00290
      SET = 1.0                                                          JUP00300
C CALL SETDAT TO RE-INITIALIZE THE VARIABLES FOR EACH AIRSPEED AND ALTIT JUP00310
      CALL SETDAT                                                        JUP00320
      SWITCH = -1.0                                                      JUP00330
C CALL PRINT TO PRINT THE OUTPUT                                         JUP00340
      CALL PRINT                                                         JUP00350
C CALL TRAJ TO CALCULATE THE SOLUTION TO THE DIFFERENTIAL EQUATIONS      JUP00360
      CALL TRAJ                                                          JUP00370
    2 CONTINUE                                                           JUP00380
    3 CONTINUE                                                           JUP00390
    4 CONTINUE                                                           JUP00400
    5 CONTINUE                                                           JUP00410
      GO TO 100                                                          JUP00420
 9999 STOP                                                               JUP00430
      END                                                                JUP00440
      SUBROUTINE PRINT                                                   JUP00450
      COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),  JUP00460
     1IREF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,         JUP00470
     2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,              JUP00480
     3RAD,  FRACT,  VE,  THETA, U, DEL, V, VXA, VYA,                     JUP00490
```

```
     4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,                JUP00500
     5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,         JUP00510
     6YO, VYO, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                     JUP00520
1000 FORMAT (' ',6X,'VKTS     ALT    DEG     TF        RANGE',4X,'IDNO'/JUP00530
    1)                                                                     JUP00540
3000 FORMAT (1H )                                                          JUP00550
5000 FORMAT (5X,F6.0,2X,F7.0,2X,F4.0,2X,F7.2,2X,F9.1,2X,I4)                JUP00560
     IF (SWITCH) 3,1,2                                                     JUP00570
   1 WRITE (6,5000) VKTS(IV),ALT(IY),DEG(IANG),T,X,IDNO                    JUP00580
     RETURN                                                                JUP00590
   2 WRITE(6,1000)                                                         JUP00600
     SWITCH = 0.0                                                          JUP00610
     RETURN                                                                JUP00620
   3 WRITE(6,3000)                                                         JUP00630
     RETURN                                                                JUP00640
     END                                                                   JUP00650
     SUBROUTINE INPUT                                                      JUP00660
     COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),     JUP00670
    1IREF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,            JUP00680
    2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,                 JUP00690
    3RAD,  FRACT,  VE,  THETA, U, DEL, V, VXA, VYA,                        JUP00700
    4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,                JUP00710
    5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,         JUP00720
    6YO, VYO, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                     JUP00730
2000 FORMAT (16G5.0)                                                       JUP00740
     READ(5,2000) KIV, (VKTS(I), I=1, KIV)                                 JUP00750
     READ (5,2000) KY, (ALT(I), I=1, KY)                                   JUP00760
     READ (5,2000) KDEG, (DEG(I), I=1, KDEG)                               JUP00770
     READ (5,2000) KIDNO, (NUMID(I), I=1, KIDNO)                           JUP00780
     RETURN                                                                JUP00790
     END                                                                   JUP00800
     SUBROUTINE SETDAT                                                     JUP00810
     COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),     JUP00820
    1IREF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,            JUP00830
    2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,                 JUP00840
    3RAD,  FRACT,  VE,  THETA, U, DEL, V, VXA, VYA,                        JUP00850
    4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,                JUP00860
    5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,         JUP00870
    6YO, VYO, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                     JUP00880
     IF (SET) 1,2,3                                                        JUP00890
   1 G= 32.174                                                             JUP00900
     RAD= 0.0174533                                                        JUP00910
     A= 0.7                                                                JUP00920
     AA= 0.5/A                                                             JUP00930
     YT = 0.0                                                              JUP00940
     VYK = -5.0                                                            JUP00950
     FRACT= 0.5                                                            JUP00960
     RETURN                                                                JUP00970
```

```
    2 DS2= 0.0                                                                    JUP00980
      CFORM1= 0.0                                                                 JUP00990
      CFORM2= 0.0                                                                 JUP01000
      DM1= 0.0                                                                    JUP01010
      DM2= 0.0                                                                    JUP01020
      DKG1= 0.0                                                                   JUP01030
      DKG2= 0.0                                                                   JUP01040
      VMUZ= 0.0                                                                   JUP01050
      VE= 0.0                                                                     JUP01060
      SL= 0.0                                                                     JUP01070
      DMAX= 5.0                                                                   JUP01080
      ITYPE= -1                                                                   JUP01090
      FN= 0.0                                                                     JUP01100
      IBOTH= 1                                                                    JUP01110
      T= 0.0                                                                      JUP01120
      RETURN                                                                      JUP01130
    3 U= VKTS(IV)*1.6878                                                          JUP01140
      DEL= ATAN2(VE,U)                                                            JUP01150
      V= SQRT(U*U+VE*VE)                                                          JUP01160
      THETA= DEG(IANG)*RAD                                                        JUP00270
      VXA= (V+VMUZ)* COS(THETA-DEL)                                               JUP01170
      VYA= (V+VMUZ)* SIN(THETA-DEL)                                               JUP01180
      RETURN                                                                      JUP01190
      END                                                                         JUP01200
      SUBROUTINE DECODE                                                           JUP01210
      COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),           JUP01220
     1IREF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,                  JUP01230
     2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,                       JUP01240
     3RAD,  FRACT,  VE,  THETA, U, DEL, V, VXA, VYA,                              JUP01250
     4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,                       JUP01260
     5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,                JUP01270
     6YO, VYC, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                            JUP01280
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23          JUP01290
     1,24,25,26,27,28), IDNO                                                      JUP01300
C WEAPON CONSTANTS FOR THE MK 43 UNRETARDED                                       JUP01310
    1 IREF= 4                                                                     JUP01320
      DKG1= 2.5506E-03                                                            JUP01330
      DTI = 3.                                                                    JUP01340
      GO TO 31                                                                    JUP01350
C WEAPON CONSTANTS FOR THE MK 57 UNRETARDED                                       JUP01360
    2 IREF= 4                                                                     JUP01370
      DKG1= 6.2994E-03                                                            JUP01380
      DTI = 3.                                                                    JUP01390
      GO TO 31                                                                    JUP01400
C WEAPON CONSTANTS FOR THE MK 61 UNRETARDED                                       JUP01410
    3 IREF= 4                                                                     JUP01420
      DKG1= 4.01E-03                                                              JUP01430
      DTI = 3.                                                                    JUP01440
```

```
      GO TO 31                                                          JUP01450
C WEAPON CONSTANTS FOR THE MK 116 WETEYE                                JUP01460
    4 IREF= 2                                                           JUP01470
      DMAX = 3.0                                                        JUP01480
      CFORM1= 3.9235E-03                                                JUP01490
      DKG1= 2.754E-03                                                   JUP01500
      DTI = 2.                                                          JUP01510
      GO TO 31                                                          JUP01520
C WEAPON CONSTANTS FOR THE MK 76 WITH LUG                               JUP01530
    5 IREF= 2                                                           JUP01540
      DMAX = 3.0                                                        JUP01550
      CFORM1= 3.9077E-03                                                JUP01560
      DKG1= 6.3648E-03                                                  JUP01570
      DTI = 1.                                                          JUP01580
      GO TO 31                                                          JUP01590
C WEAPON CONSTANTS FOR THE MK 77 FIREBOMB                               JUP01600
    6 IREF= 4                                                           JUP01610
      DMAX = 2.                                                         JUP01620
      DKG1= 0.021266                                                    JUP01630
      DTI = 1.                                                          JUP01640
      GO TO 31                                                          JUP01650
C WEAPON CONSTANTS FOR THE MK 81                                        JUP01660
    7 IREF= 1                                                           JUP01670
      CFORM1= 2.5704                                                    JUP01680
      DTI = 3.                                                          JUP01690
      GO TO 31                                                          JUP01700
C WEAPON CONSTANTS FOR THE MK 81 SNAKEYE UNRETARDED                     JUP01710
    8 IREF= 4                                                           JUP01720
      DMAX = 3.0                                                        JUP01730
      DKG1= 9.767E-03                                                   JUP01740
      DTI = 2.                                                          JUP01750
      GO TO 31                                                          JUP01760
C WEAPON CONSTANTS FOR THE MK 82 MECH FUZE                              JUP01770
    9 IREF= 1                                                           JUP01780
      CFORM1= 2.064                                                     JUP01790
      DTI = 3.                                                          JUP01800
      GO TO 31                                                          JUP01810
C WEAPON CONSTANTS FOR THE MK 82 ELEC FUZE                              JUP01820
   10 IREF= 1                                                           JUP01830
      CFORM1= 1.4932                                                    JUP01840
      DTI = 3.                                                          JUP01850
      GO TO 31                                                          JUP01860
C WEAPON CONSTANTS FOR THE MK 83 MECH FUZE                              JUP01870
   11 IREF= 1                                                           JUP01880
      CFORM1= 1.3431                                                    JUP01890
      DTI = 1.                                                          JUP01900
      GO TO 31                                                          JUP01910
C WEAPON CONSTANTS FOR THE MK83 ELEC FUZE                               JUP01920
```

```
   12 IREF= 1                                                          JUP01930
      CFORM1= 1.21                                                     JUP01940
      DTI = 3.                                                         JUP01950
      GO TO 31                                                         JUP01960
C WEAPON CONSTANTS FOR THE MK 84                                       JUP01970
   13 IREF= 1                                                          JUP01980
      CFORM1= 1.0                                                      JUP01990
      DTI = 3.                                                         JUP02000
      GO TO 31                                                         JUP02010
C WEAPON CONSTANTS FOR THE MK 117 AL                                   JUP02020
   14 IREF= 1                                                          JUP02030
      CFORM1= 3.12                                                     JUP02040
      DKG1= -1.223E-03                                                 JUP02050
      DTI = 3.                                                         JUP02060
      GO TO 31                                                         JUP02070
C WEAPON CONSTANTS FOR THE MK 86 WET SAND FILLED                       JUP02080
   15 IREF= 1                                                          JUP02090
      DMAX = 3.                                                        JUP02100
      CFORM1= 3.4972                                                   JUP02110
      DTI = 2.                                                         JUP02120
      GO TO 31                                                         JUP02130
C WEAPON CONSTANTS FOR THE MK 88 WET SAND FILLED                       JUP02140
   16 IREF=1                                                           JUP02150
      CFORM1= 1.605                                                    JUP02160
      DTI = 3.                                                         JUP02170
      GO TO 31                                                         JUP02180
C WEAPON CONSTANTS FOR THE MK 82 SNAKEYE UNRETARDED                    JUP02190
   17 IREF= 4                                                          JUP02200
      DMAX = 3.                                                        JUP02210
      DKG1= 7.329E-03                                                  JUP02220
      DTI = 1.                                                         JUP02230
      GO TO 31                                                         JUP02240
C WEAPON CONSTANTS FOR THE MK 82 SNAKEYE RETARDED                      JUP02250
   18 IREF= 1                                                          JUP02260
      ITYPE= 1                                                         JUP02270
      IBOTH= 2                                                         JUP02280
      DKG1= 7.329E-03                                                  JUP02290
      CFORM2= 1.6895E-02                                               JUP02300
      DM2= 0.38                                                        JUP02310
      DKG2= 0.17166                                                    JUP02320
      DS= 0.6617                                                       JUP02330
      SL= -0.000269                                                    JUP02340
      DTI = 2.0                                                        JUP02350
      GO TO 31                                                         JUP02360
C WEAPON CONSTANTS FOR THE SADEYE T1 = 4.0                             JUP02370
   19 IREF= 1                                                          JUP02380
      ITYPE= 1                                                         JUP02390
      IBOTH= 2                                                         JUP02400
```

```
      CFORM1= 2.0754                                                    JUP02410
      CFORM2= 0.2217                                                    JUP02420
      DS = 4.267                                                        JUP02430
      DTI = 1.5                                                         JUP02440
      GO TO 31                                                          JUP02450
C WEAPON CONSTANTS FOR THE ROCKEYE II T1 = 4.0                          JUP02460
   20 IREF= 1                                                           JUP02470
      ITYPE= 1                                                          JUP02480
      IBOTH= 2                                                          JUP02490
      CFORM1= 2.2973                                                    JUP02500
      DM1= 0.32                                                         JUP02510
      DKG1= 8.175E-03                                                   JUP02520
      CFORM2= 1.1136E-02                                                JUP02530
      DM2= 0.41                                                         JUP02540
      DKG2= 0.16885                                                     JUP02550
      DS = 4.06                                                         JUP02560
      DTI = 2.0                                                         JUP02570
      GO TO 31                                                          JUP02580
C WEAPON CONSTANTS FOR THE CBU T1 = 4.0                                 JUP02590
   21 IREF= 1                                                           JUP02600
      ITYPE= 1                                                          JUP02610
      IBOTH= 2                                                          JUP02620
      CFORM1= 2.2404                                                    JUP02630
      CFORM2= 0.1178                                                    JUP02640
      DS = 4.0                                                          JUP02650
      DTI = 1.62                                                        JUP02660
      GO TO 31                                                          JUP02670
C WEAPON CONSTANTS FOR THE MK 81 SNAKEYE RETARDED                       JUP02680
   22 IREF= 1                                                           JUP02690
      ITYPE= 1                                                          JUP02700
      IBOTH= 2                                                          JUP02710
      DKG1= 9.767E-03                                                   JUP02720
      CFORM2= 2.30625E-02                                               JUP02730
      DM2= 0.38                                                         JUP02740
      DKG2= 0.23287                                                     JUP02750
      DS= 0.679                                                         JUP02760
      SL= -0.000303                                                     JUP02770
      DTI = 1.622                                                       JUP02780
      GO TO 31                                                          JUP02790
C WEAPON CONSTANTS FOR THE GUN                                          JUP02800
   23 IREF= 3                                                           JUP02810
      DMAX = 1.5                                                        JUP02820
      CFORM1= 2.9964                                                    JUP02830
      DKG1= -0.014992                                                   JUP02840
      VMUZ= 3300.0                                                      JUP02850
      DTI = 0.5                                                         JUP02860
      GO TO 31                                                          JUP02870
C WEAPON CONSTANTS FOR THE ROCKETS                                      JUP02880
```

```
   24 IREF= 3                                                          JUP02890
      ITYPE= 2                                                         JUP02900
      CFORM1= 0.82                                                     JUP02910
      CFORM2 = 1.0                                                     JUP02920
      FN= 1746.0                                                       JUP02930
      DS = 1.4225                                                      JUP02940
      DTI = 1.                                                         JUP02950
      GO TO 31                                                         JUP02960
C WEAPON CONSTANTS FOR THE MK 43 RETARDED 0.4 SEC DELAY                JUP02970
   25 IREF= 4                                                          JUP02980
      ITYPE= 0                                                         JUP02990
      DS= 0.98                                                         JUP03000
      DKG2= 1.48                                                       JUP03010
      DTI = 0.31                                                       JUP03020
      GO TO 31                                                         JUP03030
C WEAPON CONSTANTS FOR THE MK 57 RETARDED 0.8 SEC DELAY                JUP03040
   26 IREF= 4                                                          JUP03050
      ITYPE= 0                                                         JUP03060
      DS= 0.89                                                         JUP03070
      DKG2= 2.0                                                        JUP03080
      DTI = 0.22                                                       JUP03090
      GO TO 31                                                         JUP03100
C WEAPON CONSTANTS FOR THE MK 61 RETARDED 0.6 SEC DELAY                JUP03110
   27 IREF= 4                                                          JUP03120
      ITYPE= 0                                                         JUP03130
      DS= 0.89                                                         JUP03140
      DKG2= 2.70                                                       JUP03150
      DTI = 0.1                                                        JUP03160
      GO TO 31                                                         JUP03170
C WEAPON CONSTANTS FOR THE MK 106 MOD 2                                JUP03180
   28 IREF= 2                                                          JUP03190
      ITYPE= 2                                                         JUP03200
      CFORM1= 0.1514                                                   JUP03210
      CFORM2= 0.1514                                                   JUP03220
      DS = 0.5                                                         JUP03230
      DTI = 0.8                                                        JUP03240
C SET THE REFERENCE DRAG CURVE COEFFICIENTS AND CUTS                   JUP03250
   31 GO TO (32,33,34,51), IREF                                        JUP03260
   32 CC(1,1,1)= 1.572924E-03                                          JUP03270
      CC(1,2,1)= 0.0                                                   JUP03280
      CC(1,3,1)= 0.0                                                   JUP03290
      CC(2,1,1)= 4.678409E-02                                          JUP03300
      CC(2,2,1)=-0.109711069                                           JUP03310
      CC(2,3,1)= 6.654801E-02                                          JUP03320
      CC(3,1,1)=-0.116380157                                           JUP03330
      CC(3,2,1)= 0.217643894                                           JUP03340
      CC(3,3,1)= -9.767068E-02                                         JUP03350
      CT(1,1)= 0.834                                                   JUP03360
```

```
      CT(2,1)= 0.977                                                    JUP03370
      IF (IBOTH-1) 33,51,33                                             JUP03380
   33 CC(1,1,IBOTH)= 3.53503924                                         JUP03390
      CC(1,2,IBOTH)=-3.34778216                                         JUP03400
      CC(1,3,IBOTH)= 2.87262413                                         JUP03410
      CC(2,1,IBOTH)=11.2616503                                          JUP03420
      CC(2,2,IBOTH)=-27.4162512                                         JUP03430
      CC(2,3,IBOTH)= 21.7308359                                         JUP03440
      CC(3,1,IBOTH)=-23.7915472                                         JUP03450
      CC(3,2,IBOTH)= 44.2607764                                         JUP03460
      CC(3,3,IBOTH)=-14.4996046                                         JUP03470
      CT(1,IBOTH)= 0.622                                                JUP03480
      CT(2,IBOTH)= 0.885                                                JUP03490
      GO TO 51                                                          JUP03500
   34 CC(1,1,1)=  0.104115                                              JUP03510
      CC(1,2,1)= -0.230347                                              JUP03520
      CC(1,3,1)=  0.167644                                              JUP03530
      CC(2,1,1)= -0.194037                                              JUP03540
      CC(2,2,1)=  0.401478                                              JUP03550
      CC(2,3,1)= -0.164612                                              JUP03560
      CC(3,1,1)=  7.33246E-02                                           JUP03570
      CC(3,2,1)= -2.03275E-02                                           JUP03580
      CC(3,3,1)=  2.44682E-03                                           JUP03590
      CT(1,1)= 1.032                                                    JUP03600
      CT(2,1)= 1.30                                                     JUP03610
   51 RETURN                                                            JUP03620
      END                                                               JUP03630
      SUBROUTINE TRAJ                                                   JUP03640
      COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40), JUP03650
     1IREF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,        JUP03660
     2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,             JUP03670
     3RAD,  FRACT,  VE,  THETA, U, DEL, V, VXA, VYA,                    JUP03680
     4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,             JUP03690
     5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,      JUP03700
     6YO, VYO, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                  JUP03710
C INITIALIZE THE VARIABLES FOR THE TRAJECTORY SUBROUTINE                JUP03720
      CF= CFORM1                                                        JUP03730
      DM= DM1                                                           JUP03740
      DKG= DKG1                                                         JUP03750
      MSTG= 1                                                           JUP03760
      X= 0.0                                                            JUP03770
      T= 0.0                                                            JUP03780
      VX= VXA                                                           JUP03790
      VY= VYA                                                           JUP03800
      TH= FN                                                            JUP03810
      Y= ALT(IY)                                                        JUP03820
      YA= Y                                                             JUP03830
C TYPE OF DRAG                                                          JUP03840
```

```
      IF (ITYPE) 2,1,1                                                          JUP03850
C SET STEP SIZE FOR FIRST STAGE DRAG                                            JUP03860
    1 D= DS+SL*U                                                                JUP03870
      GO TO 3                                                                   JUP03880
C COMPUTE STEP SIZE                                                             JUP03890
    2 D = DMAX                                                                  JUP03900
C CALL RUNGE KUTTA SUBROUTINE                                                   JUP03910
    3 CALL RUNGE                                                                JUP03920
      DTV = 1/G*(VY+SQRT(VY**2+2.*G*(Y)))                                       JUP03930
      D = DTI                                                                   JUP03940
      IF ((IDNO.LE.17).OR.(IDNO.EQ.23)) GO TO 4                                 JUP03950
C SET THE SECOND STAGE DRAG PARAMETERS                                          JUP03960
      MSTG = 2                                                                  JUP03970
      IF (ITYPE.EQ.2) MSTG = 1                                                  JUP03980
      DKG= DKG2                                                                 JUP03990
      DM= DM2                                                                   JUP04000
      CF = CFORM2                                                               JUP04010
      TH = 0.0                                                                  JUP04020
    4 IF (DTV - D) 5,3,3                                                        JUP04030
C SET THE STEP SIZE TO THE VACUMN DROP TIME REMAINING                           JUP04040
    5 D= DTV                                                                    JUP04050
C SET THE DRAG PARAMETERS FOR THE FINAL INTEGRATION STEP                        JUP04060
      MSTG= 2                                                                   JUP04070
      IF (ITYPE.EQ.2) MSTG = 1                                                  JUP04080
      DKG= DKG2                                                                 JUP04090
      DM= DM2                                                                   JUP04100
      TH=0.0                                                                    JUP04110
      CF= CFORM2                                                                JUP04120
C CALL RUNGE FOR THE FINAL INTEGRATION STEP                                     JUP04130
      CALL RUNGE                                                                JUP04140
      DTV = 1/G*(VY+SQRT(VY**2+2.*G*(Y)))                                       JUP04150
C UPDATE THE TIME OF FALL AND THE DOWN RANGE TRAVEL                             JUP04160
      T = T + DTV                                                               JUP04170
      X = X +DTV*VX                                                             JUP04180
      SWITCH = 0.0                                                              JUP04190
      CALL PRINT                                                                JUP04200
      RETURN                                                                    JUP04210
      END                                                                       JUP04220
      SUBROUTINE RUNGE                                                          JUP04230
      COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),         JUP04240
     1IREF, CFORM1, DKG1, DMAX,  ITYPE, IBOTH, DM1, CFORM2, DM2,                JUP04250
     2DKG2, DS,  SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,                      JUP04260
     3RAD,  FPACT,  VE,  THETA, U, DEL, V, VXA, VYA,                            JUP04270
     4CF, DM, DKG, IANG,  X, TS, T, VX, VY,  TH, Y, YA, IV,                     JUP04280
     5IY, SET, G, A, AA, YT, VYK, IDNO,SWITCH, D, CKDG, MSTG, VXO,              JUP04290
     6YO, VYC, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                          JUP04300
C INITIALIZE THE VARIABLES FOR THE RUNGE KUTTA                                  JUP04310
      AD = A*D                                                                  JUP04320
```

```
      YO= Y                                                                    JUP04330
      VXO= VX                                                                  JUP04340
      VYO= VY                                                                  JUP04350
      RHO= 2.37576E-03-Y*(6.87557E-08-Y*6.71618E-13)                           JUP04360
      CALL DERIV                                                               JUP04370
C UPDATE POSITION AND VELOCITIES                                               JUP04380
      Y= YO+AD*VY                                                              JUP04390
      RHO= 2.37576E-03-Y*(6.87557E-08-Y*6.71618E-13)
      AP1= AP2                                                                 JUP04400
      AN1= AN2                                                                 JUP04410
      VX= VXO+AD*AN1                                                           JUP04420
      VY= VYO+AD*AP1                                                           JUP04430
      CALL DERIV                                                               JUP04440
C COMPUTE TIME, POSITION AND VELOCITIES                                        JUP04450
      T= T+D                                                                   JUP04460
      X= X+D*(VXO+AA*(VX-VXO))                                                 JUP04470
      Y= YO+D*(VYO+AA*(VY-VYO))                                                JUP04480
      VX= VXO+D*(AN1+AA*(AN2-AN1))                                             JUP04490
      VY= VYO+D*(AP1+AA*(AP2-AP1))                                             JUP04500
      RETURN                                                                   JUP04510
      END                                                                      JUP04520
      SUBROUTINE DERIV                                                         JUP04530
      COMMON CT(2,2), CC(3,3,2), VKTS(20), ALT(20), DEG(20), NUMID(40),        JUP04540
     1IREF, CFORM1, DKG1, DMAX, ITYPE, IBOTH, DM1, CFORM2, DM2,                JUP04550
     2DKG2, DS, SL, FN,VMUZ, KIV, KY, KDEG, KIDNO, IIDNO,                      JUP04560
     3RAD, FFACT, VE, THETA, U, DEL, V, VXA, VYA,                              JUP04570
     4CF, DM, DKG, IANG, X, TS, T, VX, VY, TH, Y, YA, IV,                      JUP04580
     5IY, SET, G, A, AA, YT, VYK, IDNO:SWITCH, D, CKDG, MSTG, VXO,             JUP04590
     6YO, VYC, AN1, AN2, AP1, AP2, IREG, RHO, DTI, DTV                         JUP04600
C COMPUTE THE TOTAL VELOCITY AND THE MACH OF THE WEAPON                        JUP04610
      V = SQRT(VX*VX+VY*VY)                                                    JUP04620
      CM= V*(8.9544E-04+3.26E-09*Y)+DM                                         JUP04630
C DETERMINE THE REGION OF THE DRAG CURVE THAT IS APPLICABLE                    JUP04640
      IF (CM-CT(1,MSTG)) 1,1,2                                                 JUP04650
    1 IREG= 1                                                                  JUP04660
      GO TO 5                                                                  JUP04670
    2 IF (CM-CT(2,MSTG)) 3,3,4                                                 JUP04680
    3 IREG= 2                                                                  JUP04690
      GO TO 5                                                                  JUP04700
    4 IREG= 3                                                                  JUP04710
C DO INTERMEDIATE BALLISTIC CALCULATION                                        JUP04720
    5 CKDG = DKG+CF*(CC(IREG,1,MSTG)+(CC(IREG,2,MSTG)+CC(IREG,3,MSTG)*CMJUP04730
     1)*CM)                                                                    JUP04740
      HH= TH/V-RHO*CKDG*V                                                      JUP04750
      AN2= HH*VX                                                               JUP04760
      AP2= HH*VY-G                                                             JUP04770
      RETURN                                                                   JUP04780
      END                                                                      JUP04790
```

```
 DECLARE ZE BYTE, ZZ ADDRESS;
DECLARE YE BYTE,XE BYTE;
  /* FLOATING POINT ADD ROUTINE*/
ADD: PROCEDURE (XA,YA,PA);
     DECLARE (XA,YA,ZA) ADDRESS,
       PA ADDRESS, P BASED PA BYTE,
             (I,P2,Y2,DIGIT,SINE) BYTE,
             X BASED XA BYTE,
             Y BASED YA BYTE,
              (XX,YQ) ADDRESS;


  /* PROCEDURE TO LEFT JUSTIFY MANTISSA IN BINARY */
    ADJUST: PROCEDURE;
      DO I=0 TO 15;
         IF (ZZ AND 8000H)=8000H THEN RETURN;
        ZZ=SHL(ZZ,1);
        ZE=ZE-1;
    END;   RETURN;
     END ADJUST;

    XX=SHL(DOUBLE(X),8) OR X(1);
    YQ=SHL(DOUBLE(Y),8) OR Y(1);
    R2=      X(2) AND 7FH;
    Y2=      Y(2) AND 7FH;
    DIGIT=R2-Y2;
     SINE =(X(2) AND 80H) XOR (Y(2) AND 80H);
     IF (DIGIT AND 80H)<>0    THEN DIGIT=-DIGIT;
    IF DIGIT >= 16  THEN DO:
  /* VARIABLES NOT WITHIN SIGNIFICANCE RANGE */
     IF R2 > Y2 THEN DO;
      ZZ=XX; ZE=X(2); GO TO RET;
       END;
      ZZ=YQ; ZE=Y(2); GO TO RET;
       END;
     IF Y2 = R2 THEN DO;
  /* EXPONENTS EQUAL IN ABSOLUTE VALUE      */
           IF YQ>XX THEN DO;
  /* Y > X */
              ZE= Y(2);
               IF SINE < 80H THEN GO TO EXIT1;
                          GO TO EXIT2;
                    END;
          IF YQ<XX THEN DO;
  /* X > Y */
                    ZE =X(2);
                IF SINE < 80H THEN GO TO EXIT1;
       GO TO EXIT3;
```

```
                    END;
/* X = Y */
                IF SINE < 80H THEN DO;
     ZE=X(2);       GO TO EXIT1;        END;
                ZZ=0;
                    ZE    = 0;
                GO TO RET;
                    END;
   IF Y2 > R2 THEN DO;
/* Y > X */
                ZE    =     Y(2);
          XX=SHR(XX,DIGIT);
                IF SINE < 80H THEN GO TO EXIT1;
                              GO TO EXIT2;
                    END;
/* X > Y */
                ZE    =     X(2);
          YQ=SHR(YQ,DIGIT);
                IF SINE < 80H THEN GO TO EXIT1;
       GO TO EXIT3;
    EXIT1:
     ZZ=XX+YQ;
     IF CARRY THEN DO;
         ZZ=SCR(ZZ,1);
         ZE=ZE  +1;
    END;
          GO TO RET;
        EXIT2:
                ZZ=YQ-XX;
                CALL ADJUST;
       GO TO RET;
     EXIT3:
                ZZ=XX-YQ;
                CALL ADJUST;
  RET:  P=HIGH(ZZ);P(1)= LOW(ZZ);P(2)=ZE;
       RETURN;
END ADD;
 SUB: PROCEDURE (AD,BD,CD);
 DECLARE (AD,BD,CD) ADDRESS,
  A BASED AD BYTE,
  B BASED BD BYTE,
  C BASED CD BYTE;
 DECLARE BC (3) BYTE;
  BC(0)= B(0);
  BC(1)= B(1);
  BC(2)= B(2) XOR 80H;
 CALL ADD (AD,.BC,CD);
 END SUB;
```

```
/* FLOATING POINT MULTIPLY ROUTINE */
   MULT: PROCEDURE(XA,YA,PA);
     DECLARE
     XA ADDRESS, X BASED XA BYTE,
     YA ADDRESS, Y BASED YA BYTE,
     PA ADDRESS,P BASED PA BYTE,
     TQ ADDRESS;
     IF (X=0) OR (Y=0) THEN DO; ZZ=0; ZE=0; GO TO RET;  END;
     ZZ=X(1)*Y(1);
     ZZ=X(1)*Y+HIGH(ZZ);
     TQ=X*Y(1)+ZZ;
     IF CARRY THEN ZZ=X*Y+256+HIGH(TQ);
     ELSE ZZ=X*Y+HIGH(TQ);
        IF LOW(TQ)>=80H THEN ZZ=ZZ+1;
        IF ZZ<8000H THEN DO; ZZ=SHL(ZZ,1);  ZE=-1;
          IF LOW(TQ)>=0C0H THEN ZZ=ZZ+1;
     END;
     ELSE ZE=0;
    /* ADD EXPONENTS  */
     ZE=((X(2) AND 7FH) +(Y(2) AND 7FH)-64 +ZE) OR((X(2) AND 80H)
                                          XOR ( Y(2) AND 80H));
RET:  P=HIGH(ZZ);P(1)= LOW(ZZ);P(2)=ZE;
     RETURN;
   END MULT;
/* ROUTINE TO COMPARE TWO FLOATING POINT VARIABLES */
   /*    IF X<Y COMPARE=0
         X=Y COMPARE=1
         X>Y COMPARE=2    */
COMPARE: PROCEDURE (XA,YA) BYTE;
   DECLARE X BASED XA BYTE,
           Y BASED YA BYTE,
           (XA,YA,CHECK1,CHECK2) ADDRESS,
           (XP,YP) BYTE;

/* EQUAL EXPONENTS */
     XE=X(2) AND 80H;
     YE=Y(2) AND 80H;
       IF ( XE=YE ) THEN DO;
           XP=X(2) AND 7FH;
           YP=Y(2) AND 7FH;
        CHECK1=SHL(DOUBLE(X),8) OR X(1);
        CHECK2=SHL(DOUBLE(Y),8) OR Y(1);
           IF ( XE=80H ) THEN DO;
              IF ( XP<YP ) THEN RETURN 2;
               IF (XP > YP ) THEN RETURN 0;
                 IF CHECK2 < CHECK1 THEN RETURN 0;
                 IF CHECK2 > CHECK1 THEN RETURN 2;
                 RETURN 1;
```

```
            END;
            IF ( XP < YP ) THEN RETURN 0;
              IF ( XP > YP ) THEN RETURN 2;
          IF CHECK2 < CHECK1 THEN RETURN 2;
          IF CHECK2 > CHECK1 THEN RETURN 0;
          RETURN 1;
          END;
           IF ( XE = 0 ) THEN RETURN 2;
           RETURN 0;
END COMPARE;
DIV:  PROCEDURE (XA,YA,PA);
      DECLARE (XA,YA,ZA,B,XX,YQ) ADDRESS,
       PA ADDRESS,P BASED PA BYTE,
            I BYTE,
            X BASED XA BYTE,
            Y BASED YA BYTE,
          C (3) BYTE;


      XX=SHL(DOUBLE(X),8) OR X(1);
      YQ=SHL(DOUBLE(Y),8) OR Y(1);
       IF XX =0H THEN DO;
         ZZ=0H; ZE=0H; GO TO RET;
       END;
      XX=SHR(XX,1);
      YQ=SHR(YQ,1);
ENTO: ZZ=0;
      C(2)=    X(2);
           I=1;
      B=XX-YQ;
      IF NOT CARRY THEN DO;
                  IF B=0 THEN DO;
                    ZZ=8000H;
                    ZE=(((    X(2) AND 80H) XOR (    Y(2) AND 80H))
                     OR ((C(2) AND 7FH)-(    Y(2) AND 7FH)+65));
                     GO TO RET;
                   END;
          GOTO ENT2;
          END;
      XX=SHL(XX,1);
      C(2)=    (C(2) AND 7FH) - 1;
ENT1: B=XX-YQ;
      IF CARRY THEN DO;
          XX=SHL(XX,1);
          GOTO ENT3;
          END;
 /* NO CARRY, ADD ONE TO DIVIDEND TO MANTISSA AND SHIFT LEFT */
```

```
 ENT2: XX=SHL(B,1);
       ZZ=ZZ+1;
  /* CARRY SHIFT DIVIDEND MANTISSA LEFT */
 ENT3: ZZ=SHL(ZZ,1);
       I=I+1;
       IF I< 16 THEN GOTO ENT1;
  /* SET EXPONENT */
         ZE=(((    X(2) AND 80H) XOR (    Y(2) AND 80H)) OR
                   ((C(2) AND 7FH) - (    Y(2) AND 7FH)+65));
  RET:   P=HIGH(ZZ);P(1)= LOW(ZZ);P(2)=ZE;
         RETURN;
END DIV;
       SQRT:  PROCEDURE (XA,PA);
       /*  ASSUME THAT XA IS A POSITIVE REAL NUMBER  */
        DECLARE XA ADDRESS, X BASED XA BYTE;
        DECLARE PA ADDRESS, P BASED PA BYTE;
        DECLARE (C,C1,C2,C3,B,B1,B2) BYTE;
       /* INITIAL APPROXIMATION FOR THE ROOT IS
                  MANT * EXP/2  */
        DECLARE R BYTE;
        B=X;  B1= X(1);
         B2=X(2)-64; IF (B2 AND 80H) =0 THEN B2=SHR(B2,1)+64;
                           ELSE DO; B2=-B2; B2=SHR(B2,1); B2=64-B2;
                                END;
       DO R= 1 TO 4;
             CALL DIV(XA,.B,.C);
          CALL ADD(.C,.B,.B);
           B(2)=B(2)-1;
        END;
       P=B; P(1)=B(1);  P(2)=B(2);
          RETURN;
       END SQRT;
        DECLARE (T,T1,T2,Z,Z1,Z2 )BYTE;
DECLARE        (ID,TEMP,TEMP1,TEMP2,ENTRY,ENTRY1,ENTRY2) BYTE;
    FUNCTION: PROCEDURE(I,XA);
  /*   I=0      CDTABLE
       I=1      ATMOSPHERE
       I=2      COSINE
       I=3      ATAN1
       I=4 ATAN2     */
        DECLARE (E,V,D,V1,D1) (3) ADDRESS  INITIAL(
        3000H,3300H, 2D77H,
        3100H,33B7H,2E3AH,
        3200H,346EH,2EFDH,
0000H,3525H,0000H,
        0000H,35DCH, 0000H),
          I BYTE,
       LOCK (3) BYTE INITIAL (252,181,192),
```

```
  MAX (3)   BYTE INITIAL (252,181,192),
    ENTA ADDRESS,ENT BASED ENTA BYTE,
     VALA ADDRESS,VAL BASED VALA BYTE,
    DIFFA ADDRESS,DIFF BASED DIFFA BYTE,
    VAL1A ADDRESS, VAL1 BASED VAL1A BYTE,
    (LOOKI,MAXI) BYTE,
    DIFF1A ADDRESS,DIFF1 BASED DIFF1A BYTE;
   /*  INPUT  CONSISTS OF A BASED VARIABLE,  OUTPUT WILL BE
        IN GLOBAL VARIABLES T, T1, T2.  AND Z, Z1, Z2,
           FOR RHO AND INVS RESPECTIVELY    */
   DECLARE XA ADDRESS, X BASED XA BYTE;

    ENTA=E(I);  VALA=V(I);      DIFFA=D(I);
    VAL1A=V1(I);    DIFF1A=D1(I);
   LOOKI=LOOK(I);
    MAXI=MAX(I);
   ENTRY=X; ENTRY1=X(1); ENTRY2=X(2);
   IF(LOOKI=MAXI) THEN LOOKI=0;
   ID=COMPARE(.ENTRY,ENTA+LOOKI);
        IF(ID=1) THEN GOTO EXIT1;
   /*   READ UPWARDS IN THE TABLE*/
         IF(ID>1) THEN DO;
LOOP1:   IF (LOOKI=MAXI) THEN GO TO EXIT1;
           LOOKI=LOOKI+3;
  ID=COMPARE(.ENTRY,ENTA+LOOKI);
            IF(ID=1) THEN GO TO EXIT1;
             IF(ID>1) THEN GO TO LOOP1;
        LOOKI=LOOKI-3;  GO TO EXIT2;
         END;
    /*   READ DOWNWARDS IN THE TABLE    */
LOOP2:   IF(LOOKI=0) THEN GO TO EXIT1;
         LOOKI=LOOKI-3;
    ID=COMPARE(.ENTRY,ENTA+LOOKI);
           IF(ID=1) THEN GO TO EXIT1;
             IF(ID<1) THEN GOTO LOOP2;
              GOTO EXIT2;

EXIT1:  T=VAL(LOOKI);  T1=VAL(LOOKI+1);  T2= VAL(LOOKI+2);
    IF I<>1 THEN GO TO RET;
   Z=VAL1(LOOKI);  Z1=VAL1(LOOKI+1);   Z2=VAL1(LOOKI+2);  GO TO RET;

   EXIT2:  TEMP=ENT(LOOKI);  TEMP1=ENT(LOOKI+1);  TEMP2=ENT(LOOKI+2);
      /*   CHANGE THE SIGN OF TEMP    */
         TEMP2=TEMP2 XOR 80H;
     /*   ENTRY - ENT(LOOK)  */
         CALL ADD(.TEMP,.ENTRY,.TEMP);

      /*  (ENTRY - ENT(LOOK))* DIFF(LOOK)    */
```

```
      CALL MULT(.TEMP,DIFFA+LOOKI,.T);
      CALL ADD(.T,VALA+LOOKI,.T);
      /*  VAL(LOOK)+ENTRY-ENT(LOOK) *DIFF(LOOK)     */
  IF I<>1 THEN GO TO RET;
      CALL MULT(.TEMP,DIFF1A+LOOKI,.Z);
    CALL ADD(.Z,VAL1A+LOOKI,.Z);
  RET: LOCK(I)=LOOKI;
      RETURN;
 END FUNCTION;
COSSIN: PROCEDURE(XA);
   /*  THIS ROUTINE  COMPUTES  BOTH  THE  SINE  AND
       THE  COSINE  FOR  THE  GIVEN  ANGLE  XA  IN
        RADIANS .  THE  OUTPUT    VALUES  ARE
        STORED  IN  GLOBAL  VARIABLES  Z,Z1,Z2
        FOR      COSINE   T,T1,T2  FOR SINE  */
    DECLARE  (TH,TH1,TH2,THN,THN1,THN2,TMP,TMP1,TMP2) BYTE,
    (MPI2,MPI21,MPI22)    BYTE  INITIAL (0C9H,0FH,0C1H) ,
                XA  ADDRESS , X BASED  XA  BYTE;
   TH=X; TH1=X(1); TH2=X(2);
    DECLARE OUT LABEL;
    DECLARE I BYTE;
    DO I=1 TO 4;
    THN=TH;  THN1=TH1 ; THN2=TH2;
    CALL ADD (.TH,.MPI2,.TH);/*THETA - PI2 */
                /*  THN IS THBAR AND TH IS THBAR - PI/2   */
    IF(ZE AND 80H) <> 0 THEN GO TO OUT;
    END;
    RETURN;
 OUT:
       TH2=ZE AND 7FH;
    CALL FUNCTION (2,.THN);
    THN=T ;  THN1=T1;  THN2=T2;
/* THN= COS (TH )    */
    CALL FUNCTION (2,.TH);
   /*    TH= COS PI/2-TH)        */
    TH=T;    TH1=T1; TH2=T2;
    IF ((I=1)OR (I=3)) THEN DO;
           Z=THN; Z1=THN1 ;Z2=THN2;
           T=TH;  T1=TH1 ;  T2=TH2;
           IF(I=3) THEN DO;
              Z2=Z2 XOR 80H;
              T2=T2 XOR 80H;
              RETURN;
              END;
           RETURN;
           END   ;
    Z=TH;  Z1=TH1; Z2=TH2 XOR 80H;
    T=THN;  T1=THN1; T2=THN2 ;
```

```
        IF (I=4) THEN DO;
                      T2=T2 XOR 80H;
                      Z2=Z2 XOR 80H;
                      END;
        RETURN;
    END COSSIN;
 /* DECLARATION STATEMENTS FOR THE SETDAT PROCEDURE*/
DECLARE (G,RAD,A1,AA,YT,VYK,FRACT) (3) BYTE;
DECLARE (DS2,CFORM1,CFORM2,DM1,DM2,DKG1,DKG2,VMUZ,VE,SL,FN,TM,DMAX) (3) BYTE;
DECLARE (ITYPE,IBOTH,J,SET) BYTE;
DECLARE (VKTS,ALT,DEG) (60) BYTE;
DECLARE (U,DEL,TEMP1X,TEMP2X,TEMP3,TEMP4,TEMP5,TEMP6,TEMP7,V,THETA,VXA,VYA)
        (3) BYTE;
 /* DECLARATION STATEMENTS FOR THE DECODE PROCEDURE*/
DECLARE (IREF,IDNO) BYTE;
DECLARE (DTI,DS) (3) BYTE;
DECLARE (CC) (81) BYTE;
DECLARE (CT) (18) BYTE;
 /* DECLARATION STATEMENTS FOR THE TRAJ PROCEDURE */
DECLARE (CF,DM,DKG,VX,VY,TH,Y,YA,X,D,DTV) (3) BYTE;
DECLARE (SWITCH,MSTG,TABLE1) BYTE;
DECLARE (TEMP1A,TEMP2A,TEMP3A,TEMP4A,TEMP5A,TEMP6A,TEMP7A) (3) BYTE;
 /* DECLARATION STATEMENTS FOR THE RUNGE PROCEDURE */
DECLARE (AD,YO,VXO,VYO,RHO,AP1,AP2,AN1,AN2) (3) BYTE;
DECLARE (TEMP1B,TEMP2B,TEMP3B,TEMP4B,TEMP5B,TEMP6B) (3) BYTE;
 /* DECLARATION STATEMENTS FOR THE MAIN PROGRAM */
DECLARE (BEGIN,STOP) LABEL;
DECLARE (KIV,IIDNO,KIDNO,IANG,KDEG,IV,KIV,IY,KIY) BYTE;
DECLARE (NUMID) (40) BYTE;
 /* DECLARATION STATEMENTS FOR THE DERIV PROCEDURE */
DECLARE (CM,HH,CKDG) (3) BYTE;
DECLARE (IREG,BASEADDRESS) BYTE;
SETDAT: PROCEDURE;
DECLARE CONSTANTS1 (21) BYTE
     INITIAL (080H,0B2H,046H,08EH,0FAH,03BH,0B3H,033H,040H,0B6H,0DBH,040H,
              000H,000H,000H,0A0H,000H,0C3H,080H,000H,040H);
DECLARE CONSTANTS2 (6) BYTE
     INITIAL (C00H,000H,000H,0A0H,000H,043H);
DECLARE CONSTANTS3 (2) BYTE
     INITIAL (3,1);
DECLARE PI2 (3) BYTE
     INITIAL (0C9H,00FH,043H);
 /* SET THE CONSTANTS TO THEIR SELECTED VALUES */
 IF (SET=1) THEN DO J=0 TO 2 BY 1;
     G(J)= CONSTANTS1(J);
     PAD(J)= CONSTANTS1(J+3);
     A1(J)= CONSTANTS1(J+6);
     AA(J)= CONSTANTS1(J+9);
```

```
      YT(J)= CONSTANTS1(J+12);
      VYK(J)= CONSTANTS1(J+15);
      FRACT(J)= CONSTANTS1(J+18); END; ELSE
  /* SET THE VARIABLES AS ASSIGNED IN THE DECODE PROCEDURE */
  IF (SET=2) THEN DO; DO J=0 TO 2 BY 1;
      CFORM1(J)= CONSTANTS2(J);
      CFORM2(J)= CONSTANTS2(J);
      DM1(J)= CONSTANTS2(J);
      DM2(J)= CONSTANTS2(J);
      DKG1(J)= CONSTANTS2(J);
      DKG2(J)= CONSTANTS2(J);
      VMUZ(J)= CONSTANTS2(J);
      VE(J)= CONSTANTS2(J);
      SL(J)= CONSTANTS2(J);
      FN(J)= CONSTANTS2(J);
      TM(J)= CONSTANTS2(J);
      DMAX(J)= CONSTANTS2(J+3); END;
      ITYPE= CONSTANTS3(0);
      IBOTH= CONSTANTS3(1); END; ELSE
  IF (SET=3) THEN DO;
DECLARE CONVERT (3) BYTE
      INITIAL (0D8H,0C9H,041H);
  /* CONVERT KNOTS TO FEET PER SECOND      U= VKTS(IV)*1.6878      */
      CALL MULT(.CONVERT,.VKTS,.U);
  /* APPROXIMATE THE ARCTAN FUNCTION */
      CALL DIV(.VE,.U,.DEL);
  /* DETERMINE THE VELOCITY  V=U+1/2*(VE**2)/U  APPROX TO SQRT(U**2+VE**2)  */
      CALL MULT(.VE,.VE,.TEMP1X);
      CALL DIV(.TEMP1X,.U,.TEMP2X);
      CALL MULT(.TEMP2X,.FRACT,.TEMP3);
      CALL ADD(.TEMP3,.U,.V);
  /* CALCULATE THE DIVE ANGLE      THETA= DEG(IANG)*RAD      */
      CALL MULT(.RAD,.DEG,.THETA);
  /* IF THE DIVE ANGLE IS NEGATIVE, ADD IT TO 2PI */
        IF (THETA(2) AND 80H) <> 0 THEN DO;
          CALL ADD(.THETA,.PI2,.THETA);
        END;
  /* CALCULATE THE VELOCITY IN BOTH THE X AND Y DIRECTIONS  */
  /* VXA= (V+VMUZ)* COS(THETA-DEL) */
  /* VYA= (V+VMUZ)* SIN(THETA-DEL) */
      CALL SUB(.THETA,.DEL,.TEMP4);
      CALL COSSIN(.TEMP4);
        TEMP5=Z; TEMP5(1)=Z1; TEMP5(2)=Z2;
        TEMP7=T; TEMP7(1)= T1; TEMP7(2)=T2;
      CALL ADD(.V,.VMUZ,.TEMP6);
      CALL MULT(.TEMP6,.TEMP5,.VXA);
      CALL MULT(.TEMP6,.TEMP7,.VYA); END; RETURN;
END SETDAT;
```

```
DECODE: PROCEDURE;
DECLARE (START,ONE,TWO,THREE,FOUR) LABEL;
 /* DECLARE THE STARTING POSITION OF THE VARIOUS WPN COEFF */
 DECLARE IDVEC (28) ADDRESS
  INITIAL(0,13,26,39,52,65,78,91,104,117,130,143,156,169,182,195,208,221,
          254,287,320,353,386,419,452,485,518,551);
 /* DECLARE THE MUZZLE VELOCITY AND THRUST VECTOR */
DECLARE VMFN (6) BYTE
  INITIAL(0CEH,040H,04CH,0DAH,040H,04BH);
 /* DECLARE THE CC MATRIX OF DRAG COEFFICIENTS */
DECLARE CCVALUE (81) BYTE
  INITIAL(0CEH,02AH,037H,000H,000H,000H,000H,000H,0BFH,0A0H,03CH,
          0ECH,0B0H,0BDH,088H,04AH,03DH,0EEH,058H,0BDH,0DEH,0DEH,03EH,
          0C8H,007H,0BDH,0E2H,03EH,042H,0D6H,042H,0C2H,0B7H,0D9H,042H,
          0B4H,02FH,044H,0DBH,054H,0C5H,0ADH,0D8H,045H,0BEH,055H,0C5H,
          0B1H,00BH,046H,0E7H,0FCH,0C4H,0D5H,03AH,03DH,0EBH,0E0H,0BEH,
          0ABH,0AAH,03EH,0C6H,0B1H,0BEH,0CDH,08EH,03FH,0A8H,090H,0BEH,
          096H,028H,03DH,0A6H,085H,0BBH,0A0H,05AH,038H);
 /* DECLARE THE MACH CUT MATRIX CT */
DECLARE CTVALUE (18) BYTE
  INITIAL(0D5H,081H,040H,0FAH,01CH,040H,09FH,03BH,040H,0E2H,08FH,040H,
          C84H,018H,041H,0A6H,066H,041H);
 /* DECLARE THE WPNCODE MATRIX CONTAINING THE VARIABLE FOR EACH WPN */
DECLARE WPNCODE (585) BYTE
 /* WEAPON CONSTANTS FOR THE MK 43 UNRETARDED */
  INITIAL(4,000H,000H,000H,0A7H,027H,038H,0A0H,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 57 UNRETARDED */
          4,000H,000H,000H,0CEH,068H,039H,0CAH,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 61 UNRETARDED */
          4,000H,000H,000H,083H,066H,039H,0A0H,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 116 WETEYE */
          2,080H,090H,039H,0B4H,07CH,038H,0C0H,000H,042H,080H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 76 WITH LUG */
          2,080H,00CH,039H,0D0H,090H,039H,0C0H,000H,042H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 77 FIREBOMB */
          4,000H,000H,000H,0AEH,036H,03BH,080H,000H,042H,080H,000H,041H,
 /* WEAPON CONSTANTS FOR THE MK 81 */
          1,0A4H,081H,042H,000H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 81 SNAKEYE UNRETARDED */
          4,000H,000H,0C0H,0A0H,005H,03AH,0C0H,000H,042H,080H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 82 MECH FUZE */
          1,084H,018H,042H,000H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 82 ELECT FUZE */
          1,0BFH,021H,041H,000H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 83 MECH FUZE */
          1,0ABH,0EAH,041H,000H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
 /* WEAPON CONSTANTS FOR THE MK 83 ELEC FUZE */
          1,09AH,0E1H,041H,000H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
```

```
/* WEAPON CONSTANTS FOR THE MK 84 */
        1,080H,000H,041H,C00H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
/* WEAPON CONSTANTS FOR THE MK 117 AL */
        1,0C7H,0AEH,042H,0A0H,04DH,0B7H,0A0H,000H,043H,0C0H,000H,042H,
/* WEAPON CONSTANTS FOR THE MK 86 WET SAND FILLED */
        1,0DFH,0D2H,042H,000H,000H,000H,0C0H,0C0H,042H,080H,000H,042H,
/* WEAPON CONSTANTS FOR THE MK 88 WET SAND FILLED */
        1,0CDH,070H,041H,C00H,000H,000H,0A0H,000H,043H,0C0H,000H,042H,
/* WEAPON CONSTANTS FOR THE MK 82 SNAKEYE UNRETARDED */
        4,000H,000H,000H,0F0H,028H,039H,0C0H,00CH,042H,080H,000H,041H,
/* WEAPON CONSTANTS FOR THE MK 82 SNAKEYE RETARDED */
        1,000H,000H,000H,0F0H,028H,039H,000H,000H,000H,080H,000H,042H,
      1,2,000H,000H,000H,08AH,067H,03BH,0C2H,08FH,03FH,0AFH,0C7H,03EH,
          CA9H,065H,040H,08DH,008H,0B5H,
/* WEAPON CONSTANTS FOR THE SADEYE T1= 4.0 */
        1,084H,0D3H,042H,000H,000H,000H,000H,000H,000H,0C0H,000H,041H,
      1,2,000H,000H,000H,0E3H,C05H,03EH,000H,000H,000H,000H,000H,000H,
          088H,08BH,043H,C00H,000H,000H,
/* WEAPON CONSTANTS FOR THE ROCKEYE II T1= 4.0 */
        1,093H,006H,042H,085H,0F0H,03AH,000H,000H,000H,080H,000H,042H,
      1,2,0A3H,0D7H,03FH,086H,073H,03AH,0D1H,0EBH,03FH,0ACH,0E7H,03EH,
          081H,0FBH,043H,000H,000H,000H,
/* WEAPON CONSTANTS FOR THE CBU T1= 4.0 */
        1,08FH,062H,042H,000H,000H,0C0H,000H,000H,000H,0CFH,05CH,041H,
      1,2,000H,000H,000H,0F1H,041H,03DH,000H,000H,000H,000H,000H,000H,
          080H,000H,043H,000H,000H,000H,
/* WEAPON CONSTANTS FOR THE MK 81 SNAKEYE RETARDED */
        1,000H,000H,000H,0A0H,005H,03AH,000H,000H,000H,0CFH,09DH,041H,
      1,2,C00H,000H,000H,0BCH,0EDH,03BH,0C2H,08FH,03FH,0EEH,075H,03EH,
          0ADH,0D2H,040H,09EH,0DBH,0B5H,
/* WEAPON CONSTANTS FOR THE 20 MM GUN */
        3,0BFH,CC5H,042H,0F5H,0A1H,0BAH,0C0H,000H,041H,080H,000H,040H,
      3,1,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,
          000H,C00H,000H,000H,000H,000H,
/* WEAPON CONSTANTS FOR THE 5 INCH ROCKETS */
        3,0D1H,0EBH,040H,000H,000H,000H,000H,000H,000H,080H,000H,041H,
      2,1,000H,000H,000H,08CH,000H,041H,000H,C00H,000H,000H,000H,000H,
          0B6H,014H,041H,000H,000H,000H,
/* WEAPON CONSTANTS FOR THE MK 43 RETARDED 0.4 SEC DELAY */
        4,000H,C00H,000H,000H,000H,000H,000H,000H,000H,09EH,0B8H,03FH,
      0,1,000H,000H,000H,C00H,C00H,000H,000H,000H,000H,0BDH,070H,041H,
          0FAH,0E1H,040H,000H,0C0H,000H,
/* WEAPON CONSTANTS FOR THE MK 57 RETARDED 0.8 SEC DELAY */
        4,000H,000H,000H,000H,000H,000H,000H,000H,000H,0E1H,047H,03EH,
      0,1,000H,000H,000H,000H,000H,000H,000H,000H,000H,080H,000H,042H,
          0E3H,0D7H,040H,000H,C00H,000H,
/* WEAPON CONSTANTS FOR THE MK 61 RETARDED 0.6 SEC DELAY */
        4,000H,000H,000H,000H,0C0H,000H,000H,000H,C00H,0CCH,0CCH,03DH,
```

```
        0,1,0C0H,000H,000H,000H,000H,000H,000H,000H,000H,0ACH,0CCH,042H,
            0E3H,0D7H,040H,000H,0C0H,000H,
 /* WEAPON CONSTANTS FOR THE MK 106 MOD 2 */
          2,098H,008H,03EH,000H,000H,000H,000H,000H,000H,0CCH,0CCH,040H,
        2,1,0C0H,000H,000H,098H,008H,03EH,000H,000H,000H,000H,000H,000H,
            080H,000H,040H,000H,000H,000H);
 /* ASSIGN THE REFERENCE VALUE FROM THE WPNCODE */
     IREF= WPNCODE(IDVEC(IDNO-1));
 /* ASSIGN THE VARIABLES THEIR RESPECTIVE VALUES FROM THE WPNCODE */
     DO J=0 TO 2 BY 1;
      CFORM1(J)= WPNCODE(IDVEC(IDNO-1)+1+J);
      DKG1(J)= WPNCODE(IDVEC(IDNO-1)+4+J);
      DMAX(J)= WPNCODE(IDVEC(IDNO-1)+7+J);
      DTI(J)= WPNCODE(IDVEC(IDNO-1)+10+J);
     END;
 /* DECIDE IF SINGLE DRAG WEAPON AND THEN BRANCH ACCORDINGLY */
IF (IDNO <= 17) THEN GO TO START; ELSE DO;
 /* ASSIGN THE SECOND PORTION OF THE VARIABLES FOR DUAL STAGE WPNS */
      ITYPE= WPNCODE(IDVEC(IDNO-1)+13);
      IBOTH= WPNCODE(IDVEC(IDNO-1)+14);
     DO J=0 TO 2 BY 1;
      DM1(J)= WPNCODE(IDVEC(IDNO-1)+15+J);
      CFORM2(J)= WPNCODE(IDVEC(IDNO-1)+18+J);
      DM2(J)= WPNCODE(IDVEC(IDNO-1)+21+J);
      DKG2(J)= WPNCODE(IDVEC(IDNO-1)+24+J);
      DS(J)= WPNCODE(IDVEC(IDNO-1)+27+J);
      SL(J)= WPNCODE(IDVEC(IDNO-1)+30+J);
     END; END;
 /* IF IDNO=23 ASSIGN THE MUZZLE VELOCITY TO VMUZ */
IF (IDNO = 23) THEN DO J=0 TO 2 BY 1; VMUZ(J)= VMFN(J); END;
 /* IF THE IDNO=24 ASSIGN THE THRUST TO FN */
IF (IDNO = 24) THEN DO J=0 TO 2 BY 1; FN(J)= VMFN(J+3); END;
 /* BRANCH TO THE APPROPRIATE SECTION OF THE CC MATRIX */
START: IF (IREF = 1) THEN GO TO ONE;
       IF (IREF = 2) THEN GO TO TWO;
       IF (IREF = 3) THEN GO TO THREE;
       IF (IREF = 4) THEN GO TO FOUR;
ONE: DO J=0 TO 53 BY 1; CC(J)= CCVALUE(J); END;
     DO J=0 TO 11 BY 1; CT(J)= CTVALUE(J); END;
     GO TO FOUR;
TWO: DO J=0 TO 26 BY 1; CC(J)= CCVALUE(J+27); END;
     DO J=0 TO 5 BY 1; CT(J)= CTVALUE(J+6); END;
     GO TO FOUR;
THREE: DO J=0 TO 26 BY 1; CC(J)= CCVALUE(J+54); END;
       DO J=0 TO 5 BY 1; CT(J)= CTVALUE(J+12); END;
FOUR: RETURN;
END DECODE;
INPUT: PROCEDURE;
```

```
      IDNO= IDNO+1;
      VKTS= 0E1H; VKTS(1)= 000H; VKTS(2)= 049H;
      DEG= 0A0H; DEG(1)= 000H; DEG(2)= 0C4H;
      ALT= 0BBH; ALT(1)= 080H; ALT(2)= 04CH;
      KIV= 1;
      KIDNO= 1;
      KDEG= 1;
      KIY= 1;
     IF IDNO = 29 THEN KIV = 0;
END INPUT;
PRINT: PROCEDURE;
 /* DUMMY PROCEDURE TO SIMULATE THE PASSING OF INFORMATION TO THE EXECUTIVE */
     SWITCH = 1;
END PRINT;
DERIV: PROCEDURE;
DECLARE (TEMP1C,TEMP2C,TEMP3C,TEMP4C,TEMP5C) (3) BYTE;
DECLARE CONSTANTS6 (6) BYTE
     INITIAL (0EAH,0BBH,036H,0ECH,006H,024H);
DECLARE (EIGHT,NINE,TEN) LABEL;
 /* COMPUTE THE VELOCITY OF THE WEAPON  V= SQRT(VX*VX+VY*VY)  */
     CALL MULT(.VX,.VX,.TEMP1C);
     CALL MULT(.VY,.VY,.TEMP2C);
     CALL ADD(.TEMP1C,.TEMP2C,.TEMP3C);
     CALL SQRT(.TEMP3C,.V);
 /* COMPUTE THE MACH OF THE WEAPON  CM= V*(8.955E-04+3.26E-09*Y)+DM  */
     DO J=0 TO 2 BY 1;
      TEMP1C(J)= CONSTANTS6(J);
      TEMP2C(J)= CONSTANTS6(J+3);
     END;
     CALL MULT(.Y,.TEMP2C,.TEMP3C);
     CALL ADD(.TEMP3C,.TEMP1C,.TEMP4C);
     CALL MULT(.V,.TEMP4C,.TEMP5C);
     CALL ADD(.TEMP5C,.DM,.CM);
 /* DETERMINE THE REGION OF THE DRAG CURVE WHICH IS APPLICABLE */
 IF (2 = COMPARE(.CM,.CT(MSTG))) THEN GO TO EIGHT;
     IREG=0; GO TO TEN;
EIGHT:  IF (2 = COMPARE(.CM,.CT(MSTG+3))) THEN GO TO NINE;
     IREG= 9; GO TO TEN;
NINE:  IREG= 18;
 /* DO THE INTERMEDIATE BALLISTIC CALCULATIONS */
 /* CKDG=DKG+CF*(CC(IREG,1,MSTG)+(CC(IREG,2,MSTG)+CC(IREG,3,MSTG)*CM)*CM)  */
TEN:  BASEADDRESS= TABLE1+IREG;
     CALL MULT(.CM,.CC(BASEADDRESS+6),.TEMP1C);
     CALL ADD(.TEMP1C,.CC(BASEADDRESS+3),.TEMP2C);
     CALL MULT(.CM,.TEMP2C,.TEMP3C);
     CALL ADD(.TEMP3C,.CC(BASEADDRESS),.TEMP4C);
     CALL MULT(.CF,.TEMP4C,.TEMP5C);
     CALL ADD(.TEMP5C,.DKG,.CKDG);
```

```
 /* HH= TH/V-RHO*CKDG*V  */
     CALL MULT(.V,.CKDG,.TEMP1C);
     CALL MULT(.TEMP1C,.RHO,.TEMP2C);
     CALL DIV(.TH,.V,.TEMP3C);
     CALL SUB(.TEMP3C,.TEMP2C,.HH);
 /* AN2= HH*VX  */
     CALL MULT(.HH,.VX,.AN2);
 /* AP2= HH*VY-G */
     CALL MULT(.HH,.VY,.TEMP4C);
     CALL SUB(.TEMP4C,.G,.AP2);
RETURN; END DERIV;
RUNGE: PROCEDURE;
DECLARE CONSTANTS5 (9) BYTE
  INITIAL(09BH,0B2H,038H,093H,0A6H,029H,0BDH,00BH,018H);
 /* CALCULATE THE AD VALUE      AD= A*D     */
     CALL MULT(.A1,.D,.AD);
 /* ASSIGN THE VARIABLES THEIR INITIAL VALUES     */
     DO J=0 TO 2 BY 1;
      YO(J)= Y(J);
      VXO(J)= VX(J);
      VYO(J)= VY(J);
     END;
 /* CALCULATE THE AIR DENSITY     RHO=2.37E-03-Y*(6.87E-08-Y*6.71E-13) */
     DO J=0 TO 2 BY 1;
      TEMP1B(J)= CONSTANTS5(J);
      TEMP2B(J)= CONSTANTS5(J+3);
      TEMP3B(J)= CONSTANTS5(J+6);
     END;
     CALL MULT(.Y,.TEMP3B,.TEMP4B);
     CALL SUB(.TEMP2B,.TEMP4B,.TEMP5B);
     CALL MULT(.Y,.TEMP5B,.TEMP6B);
     CALL SUB(.TEMP1B,.TEMP6B,.RHO);
 /* MAKE THE FIRST CALL TO THE DERIV PROCEDURE */
CALL DERIV;
 /* UPDATE THE POSITIONS AND THE VELOCITIES     */
 /* Y= YO+AD*VY */
     CALL MULT(.AD,.VY,.TEMP1B);
     CALL ADD(.TEMP1B,.YO,.Y);
     DO J=0 TO 2 BY 1;
      AP1(J)= AP2(J);
      AN1(J)= AN2(J);
     END;
 /* VX= VXO+AD*AN1 */
     CALL MULT(.AN1,.AD,.TEMP1B);
     CALL ADD(.VXO,.TEMP1B,.VX);
 /* VY= VYO+AD*AP1 */
     CALL MULT(.AP1,.AD,.TEMP2B);
     CALL ADD(.VYO,.TEMP2B,.VY);
```

```
 /* CALCULATE THE AIR DENSITY     RHO=2.37E-03-Y*(6.87E-08-Y*6.71E-13) */
     DO J=0 TO 2 BY 1;
      TEMP1B(J)= CONSTANTS5(J);
      TEMP2B(J)= CONSTANTS5(J+3);
      TEMP3B(J)= CONSTANTS5(J+6);
     END;
     CALL MULT(.Y,.TEMP3B,.TEMP4B);
     CALL SUB(.TEMP2B,.TEMP4B,.TEMP5B);
     CALL MULT(.Y,.TEMP5B,.TEMP6B);
     CALL SUB(.TEMP1B,.TEMP6B,.RHO);
 /* MAKE THE SECOND CALL TO THE DERIV PROCEDURE     */
CALL DERIV;
 /* COMPUTE THE TIME, POSITION AND VELOCITIES     */
 /* T= T+D  */
     CALL ADD(.TM,.D,.TM);
 /* X= X+D*(VXO+AA*(VX-VXO))  */
     CALL SUB(.VX,.VXO,.TEMP2B);
     CALL MULT(.AA,.TEMP2B,.TEMP3B);
     CALL ADD(.VXO,.TEMP3B,.TEMP4B);
     CALL MULT(.TEMP4B,.D,.TEMP5B);
     CALL ADD(.X,.TEMP5B,.X);
 /* Y= YO+D*(VYO+AA*(VY-VYO))  */
     CALL SUB(.VY,.VYO,.TEMP2B);
     CALL MULT(.AA,.TEMP2B,.TEMP3B);
     CALL ADD(.VYO,.TEMP3B,.TEMP4B);
     CALL MULT(.TEMP4B,.D,.TEMP5B);
     CALL ADD(.YO,.TEMP5B,.Y);
 /* VX= VXO+D*(AN1+AA*(AN2-AN1))  */
     CALL SUB(.AN2,.AN1,.TEMP2B);
     CALL MULT(.TEMP2B,.AA,.TEMP3B);
     CALL ADD(.TEMP3B,.AN1,.TEMP4B);
     CALL MULT(.D,.TEMP4B,.TEMP5B);
     CALL ADD(.VXO,.TEMP5B,.VX);
 /* VY= VYO+D*(AP1+AA*(AP2-AP1))  */
     CALL SUB(.AP2,.AP1,.TEMP2B);
     CALL MULT(.TEMP2B,.AA,.TEMP3B);
     CALL ADD(.AP1,.TEMP3B,.TEMP4B);
     CALL MULT(.D,.TEMP4B,.TEMP5B);
     CALL ADD(.VYO,.TEMP5B,.VY);
RETURN;
END RUNGE;
TRAJ: PROCEDURE;
DECLARE (FIVE,SIX,SEVEN) LABEL;
DECLARE CONSTANTS4 (4) BYTE
     INITIAL (000H,000H,000H,0);
 /* INITIALIZE THE VARIABLES FOR THE TRAJECTORY PROCEDURE  */
     MSTG= CONSTANTS4(3);
     TABLE1= CONSTANTS4(3);
```

```
      DO J=0 TO 2 BY 1;
      CF(J)= CFORM1(J);
      DM(J)= DM1(J);
      DKG(J)= DKG1(J);
      VX(J)= VXA(J);
      VY(J)= VYA(J);
      TH(J)= FN(J);
      Y(J)= ALT(J);
      YA(J)= Y(J);
      X(J)= CONSTANTS4(J);
      TM(J)= CONSTANTS4(J);
      END;
 /* DETERMINE THE TYPE OF DRAG */
 IF (ITYPE=3) THEN GO TO FIVE;
 /* CALCULATE THE STEP SIZE  D=DS+SL*U  */
      CALL MULT(.SL,.U,.TEMP1A);
      CALL ADD(.TEMP1A,.DS,.D);
       GO TO SIX;
 /* SET THE STEP SIZE TO THE MAX ALLOWED  D= DMAX  */
 FIVE: DO J=0 TO 2 BY 1; D(J)= DMAX(J); END;
 /* CALL THE RUNGE PROCEDURE FOR THE INTEGRATION */
  SIX: CALL RUNGE;
 /* CALCULATE THE DTV VALUE  DTV= 1/G*(VY+SQRT(VY**2+2.*G*Y))  */
      CALL MULT(.G,.Y,.TEMP2A);
      TEMP3A= TEMP2A; TEMP3A(1)= TEMP2A(1); TEMP3A(2)= TEMP2A(2)+1;
      CALL MULT(.VY,.VY,.TEMP4A);
      CALL ADD(.TEMP4A,.TEMP3A,.TEMP5A);
      CALL SQRT(.TEMP5A,.TEMP6A);
      CALL ADD(.TEMP6A,.VY,.TEMP7A);
      CALL DIV(.TEMP7A,.G,.DTV);
      DO J=0 TO 2 BY 1; D(J)= DTI(J); END;
IF (IDNO <= 17) THEN GO TO SEVEN;
IF (IDNO = 23) THEN GO TO SEVEN;
 /* SET THE SECOND STAGE DRAG PARAMETERS */
      MSTG=6; TABLE1=27;
      IF (ITYPE=2) THEN DO; MSTG=0; TABLE1= 0; END;
      DO J=0 TO 2 BY 1;
       DKG(J)= DKG2(J);
       DM(J)= DM2(J);
       CF(J)= CFORM2(J);
       TH(J)= CONSTANTS4(J); END;
 /* TEST THE STEP SIZE VERSUS THE VACUMN FALL STEP SIZE  */
 SEVEN: IF (0<>COMPARE(.DTV,.D)) THEN GO TO SIX;
 /* SET THE STEP SIZE TO THE VACUMN VALUE */
      DO J=0 TO 2 BY 1;
       D(J)= DTV(J);
      END;
 /* SET THE DRAG PARAMETERS FOR THE FINAL INTEGRATION STEP */
```

```
      MSTG= 6; TABLE1= 27;
      IF (ITYPE=2) THEN DO; MSTG= 0; TABLE1=0; END;
      DO J=0 TO 2 BY 1;
       DKG(J)= DKG2(J);
       DM(J)= DM2(J);
       TH(J)= CONSTANTS4(J);
       CF(J)= CFORM2(J);
      END;
 /* CALL RUNGE FOR THE FINAL INTEGRATION  */
CALL RUNGE;
 /* CALCULATE THE DTV VALUE  DTV= 1/G*(VY+SQRT(VY**2+2.*G*Y))  */
      CALL MULT(.G,.Y,.TEMP2A);
      TEMP3A= TEMP2A; TEMP3A(1)= TEMP2A(1); TEMP3A(2)= TEMP2A(2)+1;
      CALL MULT(.VY,.VY,.TEMP4A);
      CALL ADD(.TEMP4A,.TEMP3A,.TEMP5A);
      CALL SQRT(.TEMP5A,.TEMP6A);
      CALL ADD(.VY,.TEMP6A,.TEMP7A);
      CALL DIV(.TEMP7A,.G,.DTV);
 /* UP DATE THE TIME OF FALL OF THE WEAPON  TM= TM+DTV  */
      CALL ADD(.DTV,.TM,.TM);
 /* UP-DATE THE DOWN RANGE TRAVEL OF THE WEAPON  X= X+DTV*VX  */
      CALL MULT(.DTV,.VX,.TEMP2A);
      CALL ADD(.X,.TEMP2A,.X);
 /*  SET THE SWITCH FOR THE PRINT PROCEDURE  */
      SWITCH= 0;
CALL PRINT;
RETURN;
END TRAJ;
 /* SET THE SET SWITCH TO INITIALIZE THE CONSTANTS */
      SET= 1;
CALL SETDAT;
 /* SET THE PRINT SWITCH TO PRINT THE HEADING INFORMATION  */
      SWITCH= 1;
CALL PRINT;
 /* CALL THE INPUT PROCEDURE TO UPDATE THE VALUES OF THE PROGRAM */
       IDNO=0;
BEGIN: CALL INPUT;
 /* TEST ON KIV FOR THE STOPPING OF THE PROGRAM  */
      IF (KIV=0) THEN GO TO STOP;
 /* FORM THE LOOP FOR THE NUMBER OF IDNO'S SPECIFIED  */
      DO IIDNO=1 TO KIDNO;
 /* SET THE SETDAT SWITCH TO RE-INITIALIZE THE VARIABLES FOR EACH IDNO */
      SET=2;
CALL SETDAT;
 /* CALL DECODE TO INITIALIZE THE BOMB COEFFICIENTS */
CALL DECODE;
 /* FORM THE LOOP FOR THE VARIOUS DIVE ANGLES */
      DO IANG=1 TO KDEG;
```

```
 /* FORM THE LOOP FOR THE VARIOUS AIRSPEEDS */
      DO IV=1 TO KIV;
 /* FORM THE LOOP FOR THE VARIOUS ALTITUDES */
      DO IY = 1 TO KIY;
 /* CALL THE SETDAT PROCEDURE TO RE-INITIALIZE THE VARIABLES FOR EACH ALT */
       SET=3;
CALL SETDAT;
 /* SET THE PRINT SWITCH TO PRINT THE OUTPUT */
       SWITCH= 3;
CALL PPINT;
 /* CALL TRAJ TO CALCULATE THE SOLUTION TO THE DIFFERENTIAL EQUATIONS */
CALL TRAJ;
      END;
      END;
      END;
      END;
GO TO BEGIN;
STOP: HALT;
EOF
```

LIST OF REFERENCES

1. Naval Weapon Center Technical Publication 5416, A Ballistic Trajectory Algorithm for Digital Airborne Fire Control Systems, Duke, Arthur, and others, September 1972.

2. Hildebrand, F. B., Introduction to Numerical Analysis, p. 485-495, McGraw Hill, 1956.

3. Naval Weapons Laboratory Technical Report 2741, A-6 Program Revision, Kuty, L. R., January 1974.

4. Lambert, J. D., Computational Methods in Ordinary Differential Equations, p. 114-156, Wiley, 1973.

5. Lapidus, L., Numerical Solutions of Ordinary Differential Equations, p. 39-137, Academic Press, 1971.

6. Naval Weapons Laboratory Tactical Manual Ballistic Tables, NAVAIR 01-1C-1T-1, October 1973.

7. Pease, J. A., No Drop Bomb Simulation Using Micro-Computers, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1974.

8. Syracuse University Research Corporation Technical Report 73-129, A6A-A6E/Tram, Rodems, J. D., and others, March 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chairman, Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Assoc. Professor U. R. Kodres, Code 72Kr Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
5.	Lt. Harry A. Jupin, USN 920 Miller Avenue Clairton, Pennsylvania 15025	1
6.	Burt Chase, Code KBC Naval Surface Weapon Center Dahlgren, Virginia 22448	1
7.	T. H. Zehner Grumman Aerospace Corporation Tram Building Plant 07, Department 471 Calverton, New York 11933	1
8.	Commander Naval Air Test Center System Analysis Group Weapon System Group Patuxent River, Maryland 20670 (Attn. F. Phillips)	1